

Visualisierung der internen Prozesse Künstlicher Neuronaler Netze in Virtual Reality

Visualization of the internal processes of
Artificial Neural Networks in Virtual Reality

Masterarbeit

im Studiengang Medieninformatik
zur Erlangung des akademischen Grades
Master of Science (M. Sc.)

vorgelegt von

Nadine Meißler

Matrikel-Nr.: 780645

eingereicht am: 29. Juli 2019

Fachbereich Medien

Hochschule Düsseldorf

in Kooperation mit

Deutsches Zentrum für Luft- und Raumfahrt e. V.

Simulations- und Softwaretechnik | Intelligente und Verteilte Systeme

Erstprüfer:

Prof. Dr. rer. nat. Christian Geiger

Hochschule Düsseldorf

Zweitprüfer:

Dr. Nico Hochgeschwender

Deutsches Zentrum für Luft- und Raumfahrt e. V.

Zusammenfassung

In zahlreichen Projekten werden maschinelle Lernverfahren und Künstliche Neuronale Netze eingesetzt. Neueinsteiger, Entscheider und Entwickler benötigen wahrscheinlich in Zukunft vermehrt Verfahren, mit denen die grundlegende Funktionsweise dieser Algorithmen nachvollzogen werden kann. Aufgrund der enormen Daten, neuen selbstlernenden Algorithmen und immer schnelleren Verarbeitungsgeschwindigkeiten könnten neue Übermittlungs- und Darstellungskonzepte von Vorteil sein, die eine Nachvollziehbarkeit der internen Prozesse ermöglichen. Eine Option könnte die Virtual Reality Technologie bieten, die neue Möglichkeiten sowohl hinsichtlich der Darstellung als auch der Interaktion aufzeigt.

Diese Arbeit untersucht inwieweit die Nutzung von Virtual Reality (VR) die Visualisierung eines Convolutional Neural Networks (CNN) unterstützen kann. Im Rahmen dieser Arbeit ist ein Konzept zur Visualisierung von CNNs in VR entstanden, welches als Prototyp implementiert und evaluiert wurde. Die Ergebnisse der Evaluierung zeigen, dass die Immersion durch den Einsatz von Virtual Reality Technologie eine hohe Konzentration auf die dargestellten Informationen zu ermöglichen scheint. Die Ergebnisse zeigen außerdem, dass die Anwendung als intuitiv empfunden wurde und es einen subjektiv empfundenen Wissenszuwachs gab. Der Einsatz von VR Technologie scheint zusätzlich die Motivation zur Nutzung der Anwendung zu erhöhen. Auf diesem Gebiet sollten weitere Studien folgen, um diese Effekte näher zu untersuchen, vor allem im direkten Vergleich mit ähnlichen Anwendungen an PC-Bildschirmen.

Abstract

Numerous projects use Machine Learning methods or Artificial Neural Networks. New-comers, decision-makers and developers will probably need more procedures in the future, with which the general workings of these algorithms can be understood. Due to the enormous data, new self-learning algorithms and the ever faster processing speeds new transmission and presentation concepts which allow a traceability of the internal processes could be of advantage. One option could be Virtual Reality technology, which offers new display and interaction possibilities.

This thesis examines the extent to which the use of Virtual Reality (VR) can support the visualization of a Convolutional Neural Network (CNN). Within the scope of this thesis, a concept for the visualization of CNNs in VR was developed, which was implemented as a prototype and evaluated. The results of the evaluation show that immersion through the use of Virtual Reality technology seems to allow a high concentration on the presented information. The results also show that the application was perceived as intuitive and that there was a subjectively perceived increase in knowledge. The use of VR technology also seems to increase the motivation to use the application. More studies in this area should be conducted to further investigate these effects, especially in direct comparison with similar applications on PC screens.

Inhaltsverzeichnis

ZUSAMMENFASSUNG.....	I
ABSTRACT	II
ABBILDUNGSVERZEICHNIS.....	VI
TABELLENVERZEICHNIS.....	VIII
ABKÜRZUNGSVERZEICHNIS.....	IX
ANMERKUNG	IX
1 EINLEITUNG	1
1.1 MOTIVATION.....	1
1.2 ZIELSETZUNG.....	4
1.3 AUFBAU DER ARBEIT	5
2 HINTERGRUND	6
2.1 MACHINE LEARNING.....	6
2.1.1 <i>Deep Learning</i>	6
2.1.2 <i>Künstliche Neuronale Netze</i>	7
2.1.3 <i>Convolutional Neural Networks</i>	7
2.2 VISUALISIERUNG VON KÜNSTLICHEN NEURONALEN NETZEN.....	10
2.2.1 <i>2D-Darstellung eines CNNs nach LeCun et al.</i>	10
2.2.2 <i>TensorBoard</i>	11
2.2.3 <i>ConvNetJS MNIST Demo</i>	12
2.2.4 <i>Interaktive Node-Link Visualisierung von CNNs</i>	13
2.3 VIRTUAL REALITY	15
2.3.1 <i>Definition</i>	15
2.3.2 <i>Einschränkungen von Virtual Reality Systemen</i>	17
2.3.3 <i>Einsatzgebiete von Virtual Reality Systemen</i>	19
3 KONZEPT	20
3.1 NUTZERSZENARIO.....	20
3.2 GRUNDLAGEN	20
3.3 AUFBAU DER VISUALISIERUNG	22

3.4	INTERAKTIONSTECHNIK	27
4	UMSETZUNG.....	29
4.1	SOFTWARE UND HARDWARE.....	29
4.1.1	<i>Entwicklungs-PC</i>	<i>29</i>
4.1.2	<i>HTC Vive / Vive Pro</i>	<i>29</i>
4.1.3	<i>Unity</i>	<i>30</i>
4.1.4	<i>SteamVR und SteamVR Unity Plugin.....</i>	<i>31</i>
4.1.5	<i>TensorFlow & Keras</i>	<i>32</i>
4.2	DAS CNN-MODELL	32
4.3	VISUALISIERUNG DER FILTER UND FEATURE-MAPS.....	34
4.4	IMPLEMENTIERUNG DER VR-ANWENDUNG	36
4.4.1	<i>Umsetzung der Visualisierung</i>	<i>37</i>
4.4.2	<i>Gestaltung der virtuellen Umgebung.....</i>	<i>42</i>
4.4.3	<i>Interaktionstechnik</i>	<i>43</i>
5	EVALUIERUNG.....	46
5.1	METHODIK.....	46
5.1.1	<i>Fragebogen.....</i>	<i>47</i>
5.1.2	<i>Beobachtung</i>	<i>49</i>
5.1.3	<i>Einführungstext.....</i>	<i>50</i>
5.2	VORBEREITUNG & ABLAUF DER EVALUIERUNG	50
5.3	ERGEBNISSE DER EVALUIERUNG.....	51
5.3.1	<i>Zusammensetzung des Teilnehmerkreises.....</i>	<i>53</i>
5.3.2	<i>Wissenserwerb</i>	<i>55</i>
5.3.3	<i>Einsatz von Virtual Reality</i>	<i>59</i>
5.3.4	<i>Usability der Anwendung.....</i>	<i>61</i>
5.3.5	<i>Erweiterung der Anwendung</i>	<i>66</i>
5.3.6	<i>Auffälligkeiten in Bezug auf das Vorwissen.....</i>	<i>67</i>
5.4	DISKUSSION & AUSWERTUNG.....	67
6	FAZIT & AUSBLICK	72

LITERATURVERZEICHNIS.....	74
ANHANG	81
EIDESSTATTLICHE ERKLÄRUNG	82

Abbildungsverzeichnis

Abbildung 2-1: Visuelle Darstellung des Vorgangs auf dem Convolutional-Layer (angelehnt an (Chollet, 2018, S. 166))	8
Abbildung 2-2: Beispieldarstellung der Layer in der 2D-Darstellung von LeCun et al. (angelehnt an (LeCun et al., 1998, S. 2283)).....	10
Abbildung 2-3: Ausschnitt der Network Visualization von ConvNetJS (Bildschirmaufnahme von (Karpathy, o.D.-b))	12
Abbildung 2-4: Bildschirmaufnahme der 3D-Visualisierung von Harley (Bildschirmaufnahme von (Harley, o.D.))	14
Abbildung 2-5: Detailfenster in der 3D-Visualisierung (Bildschirmaufnahme von (Harley, o.D.)).....	14
Abbildung 3-1: Übersicht über den Aufbau der Visualisierung.....	23
Abbildung 3-2: Erste Konzeptdarstellung des 3D-Modells des CNNs	24
Abbildung 3-3: Konzeptbild zur Darstellung eines Convolutional-Layers.....	25
Abbildung 3-4: Konzeptbild für kleines 3D-Modell mit Parameterdetails	27
Abbildung 4-1: Quellcode zur Generierung des CNN-Modells mit Keras	33
Abbildung 4-2: Aufbau des verwendeten CNNs, Ausgabe der Keras-Funktion <code>model.summary()</code>	33
Abbildung 4-4: Visualisierung der Filter des Convolutional-Layers und der entstehenden Feature-Maps für ein Inputbild mit der Ziffer 5	35
Abbildung 4-3: Colormap gray von matplotlib	35
Abbildung 4-5: Python-Quellcode zum Speichern der Filterbilder	36
Abbildung 4-6: Ansicht des Unity-Editors mit der Ansicht der Szene	37
Abbildung 4-7: Das 3D-Modell in der Szene im Unity-Editor	38
Abbildung 4-8: Auswahl einer Feature-Map auf dem Convolutional-Layer	40
Abbildung 4-9: Überblick über die Visualisierung zum Convolutional-Layer.....	40

Abbildung 4-10: Übersicht über die Visualisierung zum Pooling-Layer.....	41
Abbildung 4-11: Überblick über die virtuelle Umgebung im Unity-Editor.....	43
Abbildung 4-12: Interaktion mit dem 3D-Modell	44
Abbildung 5-1: VR-Erfahrung der Teilnehmer	54
Abbildung 5-2: Vorwissen der Teilnehmer	55
Abbildung 5-3: Vergleich der Wissens einschätzung zu CNNs.....	57
Abbildung 5-4: Bewertung des Einsatzes von VR und der Konzentration im Vergleich zum PC-Bildschirm	60
Abbildung 5-5: Übersicht über die durchschnittlichen Bewertungen der Usability	62
Abbildung 5-6: Bewertung der Natürlichkeit der Bewegung in der virtuellen Umgebung	63

Tabellenverzeichnis

Tabelle 2-1: Beschreibung der Parameter eines Convolutional-Layers (Inhalt nach (Chollet, 2018, S. 166-170)).....	9
Tabelle 2-2: Funktionen von TensorBoard (Inhalt nach (Google LLC, o.D.-b)).....	11
Tabelle 2-3: Textdaten der verschiedenen Layer bei der Network Visualization von ConvNetJS (Inhalt nach (Karpathy, o.D.-b)).....	13
Tabelle 2-4: PC Mindestanforderungen für eine HTC Vive Pro (Inhalt nach (HTC Corporation, 2019a))	18
Tabelle 3-1: Übersicht über die Layer-Arten und die dargestellten Inhalte	26
Tabelle 4-1: Technische Daten der HTC Vive und Vive Pro (Inhalt nach (HTC Corporation, 2019a, 2019b))	30
Tabelle 5-1: Zuordnung der Fragen des Fragebogens zu den Oberkategorien.....	52

Abkürzungsverzeichnis

CNN	Convolutional Neural Network
DLR	Deutsches Zentrum für Luft- und Raumfahrt
HMD	Head-Mounted Display
UI	User Interface
VR	Virtual Reality

Anmerkung

In der vorliegenden Arbeit wurde aus Gründen der leichteren Lesbarkeit die gewohnte männliche Sprachform verwendet. Diese inkludiert die weibliche Form und stellt keine Wertung dar.

1 Einleitung

„The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it.”

(McCarthy, Minsky, Rochester, & Shannon, 1955)

1.1 Motivation

Als Ursprung des Begriffs Künstliche Intelligenz (Artificial Intelligence) gilt die Erwähnung in einem Projektantrag für ein Forschungsprojekt zu Künstlicher Intelligenz im Sommer 1956 in Dartmouth (McCarthy et al., 1955). In dem Antrag wird als Grundlage für die weitere Forschung die Vermutung genannt, dass jeder Aspekt des Lernens oder jede andere Eigenschaft von Intelligenz so präzise beschrieben werden kann, dass eine Maschine dazu gebracht werden kann diese zu simulieren (McCarthy et al., 1955). Inzwischen stellt der Bereich des Maschinellen Lernens (Machine Learning) ein wichtiges Teilgebiet der Künstlichen Intelligenz dar. Dabei werden einem Computerprogramm keine Regeln vorgeschrieben, um ein bestimmtes Problem zu lösen, sondern es erhält eine Menge an Beispieldaten, die aus Eingabedaten und den dazugehörigen gewünschten Ausgabedaten bestehen, und generiert daraus selbst Regeln (Chollet, 2018, S. 23). Diese können nach dem Trainingsprozess auf neue Eingabedaten angewendet werden.

In dem Projektantrag von McCarthy et al. werden auch „Neuron Nets“ (1955) als Forschungsgebiet aufgeführt. Heute bilden sogenannte Künstliche Neuronale Netze (Artificial Neural Networks) ein Teilgebiet des Maschinellen Lernens. Sie bestehen aus künstlichen Neuronen und verarbeiten die Eingabedaten auf mehreren verbundenen Schichten (Layer) (Chollet, 2018, S. 27). Der Fortschritt auf dem Gebiet der Künstlichen Intelligenz und seiner Teilgebiete wurde, und wird auch heute noch, durch die exponentiell wachsende Rechnerkapazität (Moore'sches Gesetz) (Mainzer, 2016) und durch die steigende Verfügbarkeit von digitalen Massendaten gefördert (Chollet, 2018, S. 24; LeCun, Bengio, & Hinton, 2015, S. 436). Algorithmen, die Machine Learning verwenden, benötigen meistens eine sehr große Menge an digitalen Beispieldaten zum Trainieren, bevor sie Muster in den Daten erkennen und ein Problem eigenständig lösen können (Chollet, 2018, S. 24).

Heute finden beispielsweise Künstliche Neuronale Netze Verwendung bei der Spracherkennung, etwa in Sprachassistenten von Google LLC (Schalkwyk, 2019) und Siri von Apple Inc. (Siri Team, 2017). Darüber hinaus werden sie in der Bild- und Objekterkennung eingesetzt, die unter anderem bei autonomen Fahrzeugen Einsatz findet, um die Umgebung des Fahrzeuges wahrzunehmen (von Zitzewitz, 2017). Gerade bei Systemen, die Daten ähnlich zu menschlichen Sinneseindrücken verarbeiten, kommen bevorzugt Convolutional Neural Networks (CNNs) zum Einsatz, insbesondere bei Aufgaben des maschinellen Sehens (Chollet, 2018, S. 40). Diese bilden eine spezielle Art von Künstlichen Neuronalen Netzen, als deren Vorreiter Yann LeCun gilt, der 1998 ein Convolutional Neural Network Modell zur Bilderkennung namens LeNet-5 in einem Paper vorstellte (LeCun, Bottou, Bengio, & Haffner, 1998).

Durch Projekte wie AlphaGo der Alphabet Tochter DeepMind Technologies Limited hat Künstliche Intelligenz, und insbesondere Machine Learning und Deep Learning, in den letzten Jahren vermehrt auch das öffentliche Interesse geweckt. Ein Beispiel für die beschleunigte Entwicklung in diesem Bereich ist der Vergleich zwischen den Computerprogrammen AlphaGo und seinem Nachfolger AlphaGo Zero. AlphaGo besiegte im März 2016 den hochgeachteten Profispieler Lee Sedol im Brettspiel Go (Stöcker, 2017). Die signifikanten Entwicklungsfortschritte setzten sich beschleunigt fort. Der Nachfolger AlphaGo Zero, der im Oktober 2017 in einem Artikel in der Fachzeitschrift Nature beschrieben wurde (Silver, et al., 2017), lernte ohne Daten von menschlichen Go-Experten, auf deren Grundlage sein Vorgänger sich das Spiel beibrachte (Supervised Learning). AlphaGo Zero kam gänzlich ohne menschlichen Input aus und bekam lediglich die Spielregeln von Go zur Verfügung gestellt. Allein mit dieser Eingabe spielte AlphaGo Zero 4,9 Millionen Partien Go gegen sich selbst. Am Ende des Trainings schlug AlphaGo Zero seinen Vorgänger 100 zu 0 (Silver, et al., 2017).

Ebenso wurde in den letzten Jahren die Technologie von Virtual Reality Systemen weiterentwickelt. Im Jahr 2014 überraschte das Unternehmen Facebook Inc. mit dem Kauf des Virtual-Reality-Brillen-Herstellers Oculus VR, der bis dahin noch kein marktreifes Produkt entwickelt hatte (Lindner, 2014). Knapp zwei Jahre später, im März 2016, kam dann die Oculus Rift auf den Markt, seit über 20 Jahren die erste VR-Brille für Verbraucher (Janssen, 2016). Im selben Jahr folgte im April die VR-Brille HTC Vive des taiwanischen Mobilgeräte-Herstellers HTC Corporation in Zusammenarbeit mit Valve Corporation,

dem Betreiber der Steam-Plattform (Heise Online, o.D.), sowie im Oktober PlayStation VR (PS VR) von Sony Interactive Entertainment (Fauster, 2016). Zusammen mit den verschiedenen VR-Systemen kamen VR-Anwendungen auf den Markt, vor allem Spiele. Auf der Steam-Plattform, die als größte Spielevertriebsplattform der Welt gilt (Statt, 2019), sind aktuell 4332 Anwendungen verfügbar, die als VR-unterstützend gekennzeichnet sind (Stand 09.07.2019)¹. Durch diese Entwicklungen ist die Virtual Reality Technologie für eine größere Zielgruppe zugänglich geworden, sowohl im privaten als auch im wirtschaftlichen Sektor.

In zahlreichen Projekten werden maschinelle Lernverfahren und Künstliche Neuronale Netze eingesetzt. In diesen Projekten müssen die Algorithmen optimiert und Fehler aufgedeckt werden, wofür bereits Visualisierungstools wie TensorBoard (Google LLC, o.D.-c) eingesetzt werden. Ein Problem, dass sich im Moment zeigt ist, dass das Vorgehen von Algorithmen des Maschinellen Lernens, und vor allen Dingen im Bereich des Deep Learnings, schwer zu erklären und nachzuvollziehen ist. Häufig werden sie daher als „Black Box“ (Knight, 2017) bezeichnet. Gerade wenn die Entscheidungen der Algorithmen direkten Einfluss auf sensible Bereiche unserer Infrastruktur nehmen, oder den Menschen direkt betreffen, wie in der Medizin und beim autonomen Fahren (Burton, 2018), erscheint das problematisch. Die Gesellschaft für Informatik fordert in ihrer Stellungnahme zur Nationalen Strategie für Künstliche Intelligenz der Bundesregierung beispielsweise die „Sicherstellung von Transparenz, Nachvollziehbarkeit und Überprüfbarkeit von KI-Systemen“ (Gesellschaft für Informatik e. V., 2018). Besonders wichtig erscheint es hierbei außerdem, dass Neueinsteiger in das Thema, wie auch Entscheider, die grundlegende Funktionsweise der Algorithmen erkennen können. Hierbei könnten neue Übermittlungs- und Darstellungskonzepte von Vorteil sein. Eine Option könnte die Virtual Reality Technologie eröffnen, welche neue Möglichkeiten sowohl hinsichtlich der Darstellung als auch der Interaktion aufzeigen kann.

Die Masterarbeit wurde in Kooperation mit dem Deutschen Zentrum für Luft- und Raumfahrt (DLR) verfasst. Die Abteilung Intelligente und Verteilte Systeme, ist in die Einrichtung Simulations- und Softwaretechnik in Köln, Braunschweig und Berlin eingegliedert.

¹ Eigene Recherche bei Steam unter <https://store.steampowered.com/?l=german>, Es wurde nach Anwendungen gefiltert, die als VR-unterstützend markiert sind

Sie befasst sich unter anderem themenübergreifend mit Reproduzierbarkeit, Nachvollziehbarkeit und Informations- und Graphvisualisierung, auch im Bereich der intelligenten Softwaresysteme (Deutsches Zentrum für Luft- und Raumfahrt e.V., o.D.-b). Ein wachsender Anteil des Personalaufwands in den laufenden Projekten verteilt sich auf die Entwicklung von Software (Deutsches Zentrum für Luft- und Raumfahrt e. V., o.D.-a). Da mittlerweile ein Viertel der Personalressourcen dadurch gebunden sind, wurde die Notwendigkeit gesehen, in dem Bereich Simulations- und Softwaretechnik neue und innovative Verfahren zu entwickeln, um den Aufwand bei der Entwicklung und Wartung komplexer Systeme zu optimieren (Deutsches Zentrum für Luft- und Raumfahrt e. V., o.D.-a). Die Abteilung Intelligente und Verteilte Systeme arbeitet gruppenübergreifend am Verständnis, der Entwicklung und der Optimierung von komplexen Prozessen und Softwaresystemen (Deutsches Zentrum für Luft- und Raumfahrt e. V., o.D.-c). Im Bereich Visual Computing geht es unter anderem um die Visualisierung von Software, wobei in der Abteilung auch der Einsatz von Virtual Reality für solche Anwendungen untersucht wird (Deutsches Zentrum für Luft- und Raumfahrt e. V., o.D.-c). Dafür werden kontinuierlich Abschlussarbeiten in diesem Bereich vergeben, um innovative Ansätze im wissenschaftlichen Umfeld zu betrachten.

1.2 Zielsetzung

Die Vorgabe des DLR war es, eine mögliche Visualisierung von Künstlichen Neuronalen Netzen in VR zu untersuchen. Besonders interessant, und damit untersuchungswert, erscheinen die CNNs, da sie in zahlreichen Machine Learning Projekten eingesetzt werden und eine besondere Art von Künstlichen Neuronalen Netzen darstellen.

Die Zielsetzung dieser Arbeit war, zu untersuchen, inwieweit der Einsatz von VR Technologie bei der Visualisierung einen Mehrwert schaffen kann bzw. neue Visualisierungsmöglichkeiten bietet, um CNNs besser zu verstehen. Dazu sollte ein Konzept für eine interaktive Visualisierung eines CNNs in VR entwickelt und als Prototyp implementiert und evaluiert werden. Durch die Visualisierung soll es Nicht-Experten auf dem Gebiet ermöglicht werden ein grundlegendes Verständnis der Funktionsweise von CNNs zu erlangen.

Die Implementierung sollte in Unity erfolgen, da es eine der bevorzugten Entwicklungsumgebungen in der Abteilung des DLR ist. Damit wird eine Weiterentwicklung des Systems ermöglicht.

1.3 Aufbau der Arbeit

Diese Arbeit gliedert sich in sechs Kapitel.

Nach der Einleitung wird zunächst der Hintergrund zu der Arbeit beschrieben. Dazu werden die Begriffe Machine Learning, Deep Learning, Künstliche Neuronale Netze und Convolutional Neural Networks erläutert und die Visualisierungen vorgestellt, welche das Konzept beeinflusst haben. Anschließend wird Virtual Reality definiert und Anwendungsgebiete für Virtual Reality Anwendungen vorgestellt.

Darauf folgt die Beschreibung des Konzepts, das für die Visualisierung entwickelt wurde, und die Erläuterung der Umsetzung des Konzepts als Prototyp. Anschließend werden die Vorbereitung, der Ablauf und die Ergebnisse der Evaluierung des Prototyps beschrieben.

Die Arbeit schließt ab, indem die Ergebnisse dieser Arbeit zusammengefasst und ein Fazit sowie ein Ausblick für mögliche weitere Arbeiten gegeben werden.

2 Hintergrund

In diesem Kapitel werden die Hintergrundthemen für diese Arbeit beschrieben. In Abschnitt 2.1 wird eine kurze Einführung in den Themenbereich Machine Learning gegeben, in Abschnitt 2.2 folgt eine Beschreibung von ausgewählten Visualisierungen von Künstlichen Neuronalen Netzen und in Abschnitt 2.3 wird die Virtual Reality Technologie erläutert.

2.1 Machine Learning

Machine Learning oder Maschinelles Lernen ist ein Teilbereich der Künstlichen Intelligenz. Die zentrale Fragestellung ist dabei, ob Computer selbst Regeln erlernen können, um eine Aufgabe zu erledigen, statt diese Regeln von Menschen vorgegeben zu bekommen (Chollet, 2018, S. 23). Das Grundprinzip des Machine Learnings besteht darin, dass ein Computer Beispieldaten und die dazugehörigen Ausgaben bekommt und daraus Regeln entwickelt, die nach einem Trainingsprozess auf neue Daten angewendet werden können, um ausgehend davon Ausgaben zu generieren (Chollet, 2018, S. 23). Es steht damit im Kontrast zur klassischen Programmierung, wobei die Regeln zur Lösung einer Aufgabe durch ein Programm von Menschen vorgegeben werden (Chollet, 2018, S. 23). Dadurch entfällt die Definierung eines klaren Regelwerks, um einen Computer eine bestimmte Aufgabe lösen zu lassen (Chollet, 2018, S. 23), wodurch sich die Möglichkeit ergibt auch Aufgaben an Computer zu übertragen, für die es keine von Menschen definierten Regeln gibt. Chollet bezeichnet Machine Learning daher als „neues Programmierparadigma“ (2018, S. 23).

Die Aufgabe eines Machine-Learning-Modells ist es, Eingabedaten in passende Ausgaben umzuwandeln. Der Fokus eines Machine-Learning-Modells liegt also auf der Umwandlung von Daten. Chollet beschreibt, dass „sinnvolle Repräsentationen der gegebenen Eingabedaten erlernt werden [müssen] – Repräsentationen, die uns der zu erwartenden Ausgabe näherbringen.“ (Chollet, 2018, S. 25)

2.1.1 Deep Learning

Deep Learning ist ein Teilbereich des Machine Learnings (Chollet, 2018, S. 27). Beim Deep Learning werden aufeinanderfolgende Layer (Schichten) erlernt, die mit

zunehmender Tiefe immer sinnvollere Repräsentationen der Eingabedaten bis hin zur Ausgabe liefern sollen (Chollet, 2018, S. 27). Die Tiefe des Modells entspricht dabei der Anzahl der aufeinanderfolgenden Layer, wobei jeder Layer durch das Training des Modells erlernt wird (Chollet, 2018, S. 27).

2.1.2 Künstliche Neuronale Netze

Beim Deep Learning werden meistens sogenannte Künstliche Neuronale Netze (im Folgenden auch Neuronale Netze genannt) eingesetzt, die aus aufeinanderfolgenden Layern bestehen (Chollet, 2018, S. 27). Jeder dieser Layer besteht dabei aus Neuronen, die normalerweise jeweils mit jedem Neuron des nachfolgenden Layers verbunden sind (Rashid, 2017, S. 36). Diese Verbindungen haben Gewichtungen, durch die das Signal, welches von Neuron zu Neuron gesendet wird, verstärkt oder abgeschwächt werden kann (Rashid, 2017, S. 37). Die Gewichte sind die Parameter eines Künstlichen Neuronalen Netzes, die während des Trainingsprozesses erlernt werden (Rashid, 2017, S. 38). Das Signal, das ein Neuron sendet, wird dabei durch die Eingabe der Eingangssignale des Neurons in eine sogenannte Aktivierungsfunktion berechnet (Rashid, 2017, S. 40).

2.1.3 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) sind eine spezielle Art von Künstlichen Neuronalen Netzen. Sie werden vor allem im Bereich des maschinellen Sehens angewendet, sind aber auch für andere Aufgaben geeignet, die sich auf Sinneswahrnehmungen beziehen (Chollet, 2018, S. 40). LeCun, Bottou, Bengio und Haffner beschreiben die Eigenschaften, durch die grundlegende Vorteile entstehen, die ein CNN beispielsweise bei der Bildererkennung im Gegensatz zu klassischen Neuronalen Netzen vorweisen kann: „local receptive fields, shared weights [...] and sub-sampling“ (1998, S. 2283).

Local receptive fields

Bei einem CNN ist ein Neuron eines Layers nicht mit allen Neuronen des vorherigen Layers verknüpft, sondern nur mit einem Satz von Neuronen, die sich in einem festgelegten Bereich befinden, dem rezeptiven Feld des Neurons (LeCun et al., 1998, S. 2283). Dadurch können die Neuronen elementare visuelle Merkmale (Features) extrahieren, wie Ecken und Kanten (LeCun et al., 1998, S. 2283). In den nachfolgenden Layern werden die so

extrahierten Merkmale kombiniert, um höherwertige Merkmale zu erkennen (LeCun et al., 1998, S. 2283).

Shared weights

Shared weights werden eingesetzt, um dasselbe Merkmal an unterschiedlichen Positionen im Bild erkennen zu können und das CNN resistenter für Transformationen des Bildes zu machen, die die Position eines Merkmals im Bild verändern (LeCun et al., 1998, S. 2283). Dazu bekommen Neuronen, die mit ihren rezeptiven Feldern unterschiedliche Bereiche des Bildes abdecken, dieselben Gewichte zugewiesen (LeCun et al., 1998, S. 2283). Diese Gewichte sind in sogenannten Filtern zusammengefasst. Die Größe der Filter kann eingestellt werden und entspricht der Größe des rezeptiven Feldes, welches die Neuronen betrachten (Chollet, 2018, S. 166). Die Filter stellen dabei Merkmale dar, auf die das Bild untersucht wird (Chollet, 2018, S. 166). Die Gewichte, aus denen sich die Filter zusammensetzen, und somit die Merkmale, werden vom CNN während des Trainings gelernt (Chollet, 2018, S. 166). Die Neuronen, die sich einen Satz an Gewichten „teilen“, sind auf einem Layer in einer Fläche angeordnet (LeCun et al., 1998, S. 2283). Der gesammelte Output der Neuronen dieser Fläche wird als Feature-Map bezeichnet (LeCun et al., 1998, S. 2283) und stellt das Vorkommen des Filter-Merkmals an verschiedenen Stellen im Inputbild dar (Chollet, 2018, S. 166). Abbildung 2-1 stellt das Vorgehen auf einem Convolutional-Layer dar.

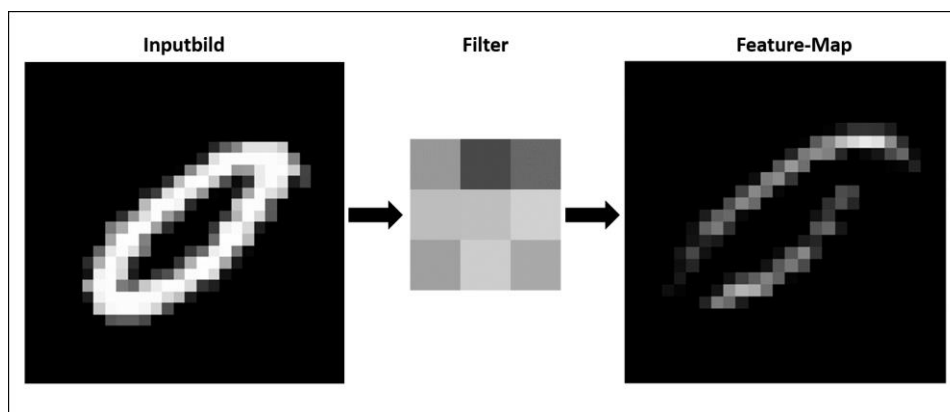


Abbildung 2-1: Visuelle Darstellung des Vorgangs auf dem Convolutional-Layer (angelehnt an (Chollet, 2018, S. 166))

Ein Convolutional-Layer besteht aus mehreren Feature-Maps, somit können an jeder Position im Bild mehrere Merkmale extrahiert werden (LeCun et al., 1998, S. 2283). Der erste Convolutional-Layer in einem CNN extrahiert die Merkmale des Inputbildes, die

folgenden Convolutional-Layer extrahieren Merkmale aus den Feature-Maps, die jeweils auf dem vorherigen Layer entstanden sind.

Eine Auflistung und Beschreibung der Parameter, die bei der Erstellung eines Convolutional-Layers nach (Chollet, 2018, S. 166-169) eingestellt werden können, ist in Tabelle 2-1 dargestellt.

Parameterbezeichnung	Beschreibung
Aktivierungsfunktion	Aktivierungsfunktion der Neuronen des Convolutional-Layers
Filtergröße	Größe des rezeptiven Feldes, das jeweils mit einem Neuron verbunden ist
Filteranzahl	Anzahl der Filter mit unterschiedlichen Gewichten, die auf einem Convolutional-Layer eingesetzt werden (entspricht der Anzahl der Feature-Maps, die auf dem Layer entstehen)
Stride	Schrittweite zwischen den rezeptiven Feldern der Neuronen
Padding	Kann genutzt werden, um die Größe der Eingabe bei der Ausgabe beizubehalten, indem die Eingabe mit Nullwerten aufgefüllt wird.

Tabelle 2-1: Beschreibung der Parameter eines Convolutional-Layers (Inhalt nach (Chollet, 2018, S. 166-170))

Sub-sampling

Sobald ein Merkmal entdeckt wurde, ist vor allem seine ungefähre Position in Relation zu anderen Merkmalen relevant, seine genaue Position ist jedoch weniger wichtig (LeCun et al., 1998, S. 2283). Die exakte Position der Merkmale kann für die Bilderkennung potenziell schädlich sein, da sie sich bei verschiedenen Darstellungen unterscheiden kann (LeCun et al., 1998, S. 2283). Um die Genauigkeit, mit der die Position in einer Feature-Map festgehalten wird, zu reduzieren, kann die räumliche Auflösung der Feature-Map reduziert werden (LeCun et al., 1998, S. 2283). Dies geschieht auf den sogenannten Sub-Sampling- oder Pooling-Layern. Bei dem Max-Pooling-Verfahren werden die Feature-Maps verkleinert, indem aus einem Feld mit einstellbarer Größe (meist 2 x 2) nur der maximale Wert übernommen wird (Chollet, 2018, S. 170). Das Max-Pooling-Verfahren eignet sich laut (Chollet, 2018, S. 171-172) dafür besonders, da das maximale Vorkommen von Merkmalen mehr Aussagekraft besitzt, als beispielsweise das durchschnittliche Vorkommen auf dem das Mean-Pooling basiert. Das Max-Pooling-Verfahren ist fest

einprogrammiert und wird während des Trainingsprozesses nicht verändert (Chollet, 2018, S. 170).

2.2 Visualisierung von Künstlichen Neuronalen Netzen

Es existieren mehrere Ansätze, um Künstliche Neuronale Netze zu visualisieren. In diesem Abschnitt werden jene aufgezählt, die für die Entwicklung des Konzepts untersucht wurden.

2.2.1 2D-Darstellung eines CNNs nach LeCun et al.

Um den Aufbau eines Convolutional Neural Networks darzustellen, wird beispielsweise von LeCun et al. (1998) für LeNet-5 eine 2D-Darstellung genutzt, welche die verschiedenen Layer des Modells und ihre Abfolge und Dimensionen anzeigt. Für die Convolutional- und Pooling- bzw. Subsampling-Layer werden dabei pro Layer die Anzahl und Größe der Feature-Maps angezeigt, zusammen mit aufeinandergestapelten Vierecken, welche die Feature-Maps darstellen. Für die Fully-Connected-Layer wird die Anzahl der Neuronen angegeben. Zwischen den Layern wird die Art der Operation angeführt, die abläuft, bzw. die Art der Verbindung, also „Convolutions“, „Subsampling“, „Full connection“ oder „Gaussian connections“ (LeCun et al., 1998, S. 2283). Eine Beispieldarstellung für Convolutional-, Subsampling- und Fully-Connected-Layer kann Abbildung 2-2 entnommen werden.

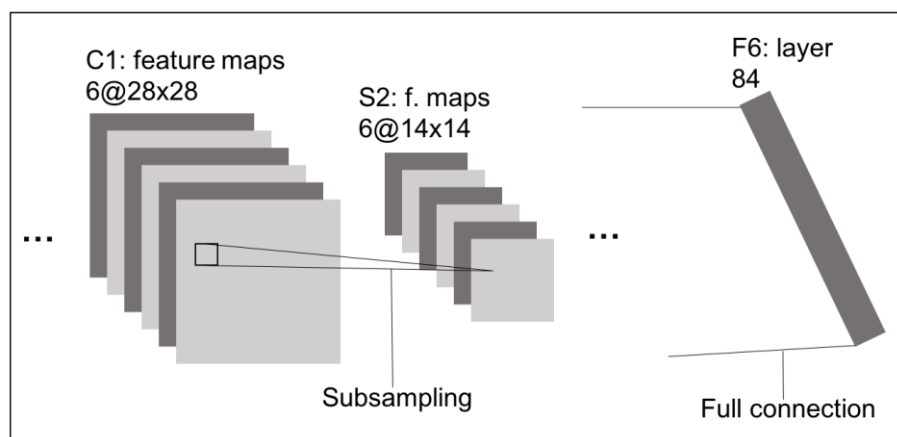


Abbildung 2-2: Beispieldarstellung der Layer in der 2D-Darstellung von LeCun et al. (angelehnt an (LeCun et al., 1998, S. 2283))

2.2.2 TensorBoard

TensorBoard ist ein Visualisierungstool, welches zum Framework TensorFlow gehört (siehe Abschnitt 4.1.5 TensorFlow & Keras), mit dem unter anderem Machine-Learning-Modelle implementiert werden können. Es besteht aus einer Sammlung von Webanwendungen, die zum Inspizieren und Verstehen von TensorFlow Graphen und deren Ausführung dienen (Google LLC, o.D.-b). Im TensorBoard Guide wird als Ziel der Visualisierungen genannt, TensorFlow Programme zu verstehen, zu debuggen und zu optimieren, wobei der Schwerpunkt auf der Visualisierung des Lernprozesses liegt (Google LLC, o.D.-c). Um das Visualisierungstool nutzen zu können, müssen die Daten, die visualisiert werden sollen, im Quellcode des TensorFlow-Modells explizit gespeichert werden (Google LLC, o.D.-c). TensorBoard bietet dann verschiedene Funktionen zur Visualisierung der Daten an, die in Tabelle 2-2 dargestellt sind.

Scalar Dashboard	Visualisierung von skalaren Statistiken, die sich über die Zeit des Trainings verändern, wie der Loss oder die Accuracy eines Modells
Histogram Dashboard	Zeigt an, wie sich die statistische Verteilung eines Tensors im Laufe der Zeit verändert hat, z. B. die Gewichte eines Convolutional-Layers
Distribution Dashboard	Basiert auf den gleichen Daten, wie das Histogram Dashboard und erweitert die Möglichkeiten zur Visualisierung
Image, Audio und Text Dashboard	Zeigt Bilder, abspielbare Audiodateien und Texte, die im Quellcode abgespeichert wurden
Graph Explorer	Visualisiert den TensorFlow Graphen
Embedding Projector	Für interaktive Visualisierungen und die Analyse von hochdimensionalen Daten

Tabelle 2-2: Funktionen von TensorBoard (Inhalt nach (Google LLC, o.D.-b))

Es können außerdem die Werte für verschiedene Parametereinstellungen verglichen werden, also beispielsweise, wie sich die *Accuracy* des Modells verändert, wenn zwei statt eines Fully-Connected-Layer genutzt werden (Google Developers, 2017).

2.2.3 ConvNetJS MNIST Demo

ConvNetJS ist eine Javascript Bibliothek zum Trainieren von Deep-Learning-Modellen in einem Browser. Sie wurde von Andrej Karpathy entwickelt, um Studenten und anderen Interessierten das Experimentieren und Beobachten des Trainingsprozesses von Neuronalen Netzen im Browser ohne weitere Software zu ermöglichen (Karpathy, ConvNetJS: Deep Learning in your browser, o.D.-a). Es gibt Demo-Versionen für verschiedene Modelle, unter anderem für ein Convolutional Neural Network, welches mit dem MNIST Datensatz, bestehend aus Bildern von handschriftlichen Ziffern von 0-9, trainiert wird (Karpathy, o.D.-b). Die Trainingsparameter, wie Learning Rate, Momentum und Batch Size sowie der Aufbau des Netzes können angepasst werden. Das Netz wird dann trainiert, wobei im Abschnitt „Network Visualization“ der Trainingsvorgang unterteilt in die Layer verfolgt werden kann (siehe Abbildung 2-3).

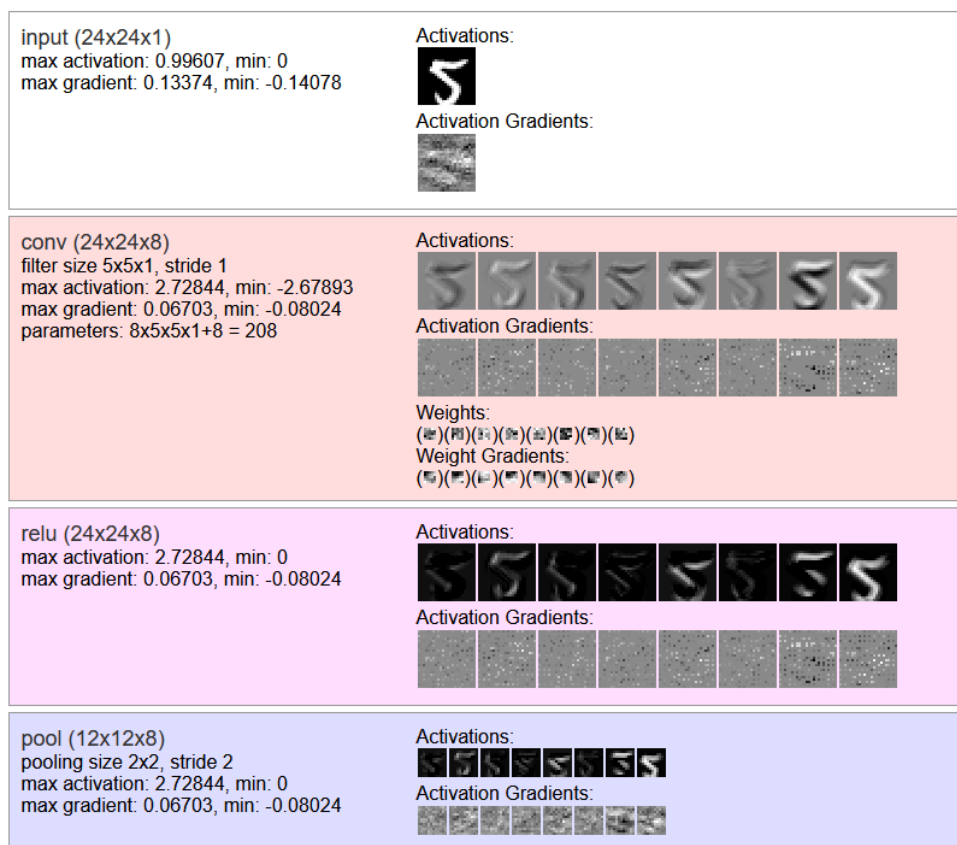


Abbildung 2-3: Ausschnitt der Network Visualization von ConvNetJS (Bildschirmaufnahme von (Karpathy, o.D.-b))

Zu jedem Layer werden Daten, wie die Dimensionen im Format Höhe x Breite x Anzahl sowie minimale und maximale Aktivierung und Gradient, angezeigt. Zusätzliche Angaben

variieren je nach Layer und können Tabelle 2-3 entnommen werden. Neben diesen Daten, die durch Text dargestellt sind, wird eine Visualisierung der Aktivierungen und der Aktivierungsgradienten angezeigt. Bei dem Convolutional-, Pooling- und ReLu-Layer entsprechen die Visualisierungen der Aktivierungen den Feature-Maps. Auf dem Convolutional-Layer werden außerdem zusätzlich Visualisierungen der Gewichte und der Gewichtsgradienten angezeigt. Jeder Layer Typ hat eine andere farbliche Hinterlegung.

Layer Name	Angezeigte Details
Input, Relu & Softmax	Maximale & minimale Aktivierung Maximaler & minimaler Gradient
Conv	Maximale & minimale Aktivierung Maximaler & minimaler Gradient Filtergröße, Stride, Anzahl der Parameter
Pool	Maximale & minimale Aktivierung Maximaler & minimaler Gradient Poolinggröße, Stride
Fc	Maximale & minimale Aktivierung Maximaler & minimaler Gradient Anzahl der Parameter/Neuronen

Tabelle 2-3: Textdaten der verschiedenen Layer bei der Network Visualization von ConvNetJS (Inhalt nach (Karpathy, o.D.-b))

2.2.4 Interaktive Node-Link Visualisierung von CNNs

Adam W. Harley (2015) hat in seinem Paper eine interaktive 3D-Visualisierung für Künstliche Neuronale Netze vorgestellt, die ein Convolutional Neural Network darstellt, welches Ziffern klassifizieren kann. Seine Visualisierung stellt im Gegensatz zu denen in Abschnitt 2.2.2 und Abschnitt 2.2.3 vorgestellten Visualisierungen nicht den Trainingsprozess des Netzes dar, sondern die Inputverarbeitung eines bereits trainierten Netzes. Eine Implementierung der Anwendung kann unter (Harley, o.D.) im Browser getestet werden (siehe Abbildung 2-4, S. 14). Als Ziel seiner Visualisierung gibt Harley (2015, S. 867) an, die Größe und Komplexität eines Netzes illustrieren zu wollen und dem Nutzer die Möglichkeit zum Experimentieren zu geben.

Der Nutzer kann eine eigene Ziffer zeichnen und die Anwendung stellt für jede Feature-Map die einzelnen Aktivierungen der Neuronen dar. Dabei wird farblich abstrahiert, wie

hoch die Aktivierung ist. Bei der Auswahl einer Aktivierung bzw. eines Neurons, werden die Verbindungen zum vorherigen Layer angezeigt, also welcher Input des vorherigen Layers zu der Aktivierung führt.

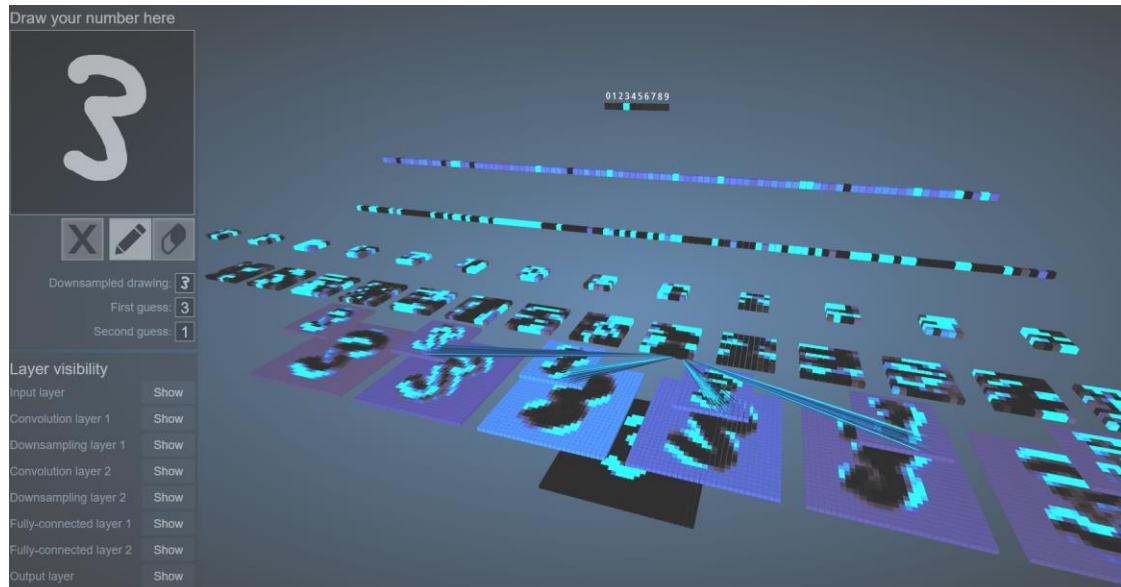


Abbildung 2-4: Bildschirmaufnahme der 3D-Visualisierung von Harley (Bildschirmaufnahme von (Harley, o.D.))

Zusätzlich werden weitere Details, wie eine Gegenüberstellung des Inputbereiches und des Filters und der Output des Neurons eingeblendet (siehe Abbildung 2-5). Für die Convolutional- und Pooling-Layer werden die Aktivierungen im Viereck angeordnet, sodass sie eine Feature-Map-Fläche bilden. Für die Fully-Connected- und Output-Layer werden die einzelnen Neuronen in einer Reihe angezeigt. Der Nutzer kann die Visualisierung in alle Richtungen drehen und zoomen. Des Weiteren besteht die Möglichkeit einen oder mehrere Layer auszublenden.

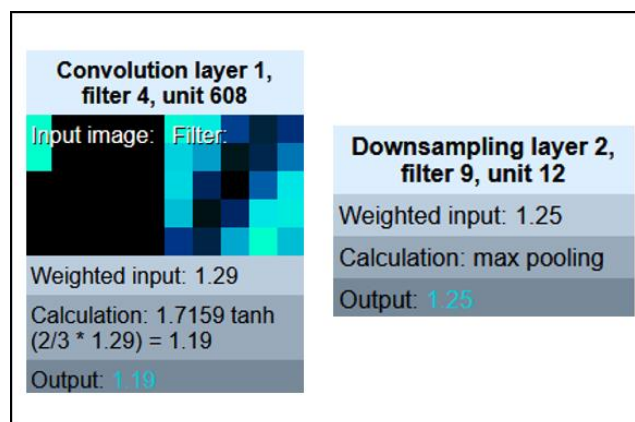


Abbildung 2-5: Detailfenster in der 3D-Visualisierung (Bildschirmaufnahme von (Harley, o.D.))

2.3 Virtual Reality

In diesem Abschnitt wird der Begriff Virtual Reality, wie er im Kontext dieser Arbeit verstanden wird, erläutert und Einschränkungen und Einsatzgebiete von Virtual Reality Systemen beschrieben.

2.3.1 Definition

Für den Begriff Virtual Reality, bzw. Virtuelle Realität, existiert keine einheitliche Definition, über die ausschlaggebenden Merkmale von Virtual Reality gibt es jedoch einen weitgehenden Konsens (Dörner, Jung, Grimm, Broll, & Göbel, 2013, S. 12). Ein Schwerpunkt in der Abgrenzung von VR-Systemen zu „traditionellen Mensch-Computer-Schnittstellen“ (Dörner et al., 2013, S. 12) ist die Technologie, die verwendet wird. Auffallend sind dabei beispielsweise die häufig von VR-Systemen verwendeten Head-Mounted Displays (HMDs), die mit Helmen direkt am Kopf des Nutzers angebracht sind und als Ausgabegerät dienen (Dörner et al., 2013, S. 12-13). Psotka nannte HMDs im Jahr 1995 „The core technology that makes immersive Virtual Reality possible“ (S. 407). Eine Version dieser Technologie wurde auch im Rahmen dieser Arbeit verwendet.

Ein anderer Ansatz ist der Einsatz einer sogenannten CAVE, wie sie an der RWTH Aachen verwendet wird und die ebenfalls zur Darstellung immersiver, virtueller Umgebungen dient (IT Center der RWTH Aachen, 2019). Die aixCAVE in Aachen besteht aus fünf Projektionsflächen und hat eine Grundfläche von mehr als 5x5 Quadratmetern, bei einer Höhe von über drei Metern, womit sie die größte CAVE weltweit ist (IT Center der RWTH Aachen, 2019).

Laut Dörner et al. (2013, S. 13) sollte bei einer Definition von VR jedoch darauf geachtet werden, sich nicht zu sehr auf bestimmte Technologien festzulegen, damit die Definition bei zukünftigen technologischen Entwicklungen nicht ihre Gültigkeit verliert. Die Definition von Cruz-Neira (1993, zitiert nach Dörner et al., 2013, S. 13) scheint daher geeignet, um das Verständnis des Begriffs Virtual Reality im Rahmen dieser Arbeit zu erläutern. Sie verbindet die Anforderungen an die virtuelle Umgebung mit den Technologien, die benötigt werden, um diese zu generieren, ohne dabei jedoch diese Technologien durch eine genaue Spezifizierung einzuschränken:

„Virtual Reality refers to immersive, interactive, multi-sensory, viewer-centered, three-dimensional computer generated environments and the combination of technologies required to build these environments.”

(Cruz-Neira, 1993, zitiert nach Dörner et al., 2013, S. 13)

Als erste Eigenschaft von Virtual Reality wird in dieser Definition die Immersion genannt. Auch Psotka legt den Fokus auf die Immersion und nutzt dafür den Begriff „Immersive VR“ (1995, S. 406), um eine Abgrenzung zu Systemen zu schaffen, die auch als VR-Systeme bezeichnet werden, jedoch nicht immersiv sind. Als Hauptmerkmal immersiver VR nennt er die Platzierung einer Person in eine simulierte Umgebung, die zu einem gewissen Grad aussieht und sich anfühlt wie die reale Welt, wobei die Person ein Gefühl der Selbstlokalisierung hat (Psotka, 1995, S. 406). Die virtuelle Umgebung kann dabei durch Kopf- und Augenbewegungen erkundet werden, wobei die Person das Gefühl hat, von dem virtuellen Raum umgeben zu sein und mit den Objekten darin interagieren kann (Psotka, 1995, S. 406). Bricken (1990, zitiert nach Dörner et al., 2013, S. 13) äußert sich ähnlich und nennt als primäre definierende Charakteristik von VR die Einbeziehung des Nutzers. Er meint: „VR places the participant inside information.“ (Bricken, 1990, zitiert nach Dörner et al., 2013, S. 13) In Bezug auf VR sagt Psotka außerdem: „it provides a fundamentally different mode of communication between computer and person“ (Psotka, 1995, S. 410).

Ein weiterer Begriff, der sich an den Begriff der Immersion anschließt, ist die Präsenz. Witmer und Singer (1998, S. 225) definieren Präsenz als das subjektive Gefühl an einem Ort oder in einer Umgebung zu sein, obwohl man sich physikalisch gesehen woanders befindet. Somit beschreibt die Präsenz in Bezug auf VR, das Gefühl sich in der virtuellen Umgebung zu befinden, statt in der realen Umgebung und drückt damit eine starke Immersion durch die VR-Anwendung aus.

Die Generierung der virtuellen Umgebung in Echtzeit ist ein weiteres Merkmal von Virtual Reality, das zum Teil die Immersion ermöglicht. Dadurch wird es möglich, dass sich der Nutzer in der virtuellen Umgebung umschauen und bewegen kann und diese dabei „als kontinuierlich und natürlich wahrnimmt“ (Dörner et al., 2013, S. 6-7). Dörner et al. (2013, S. 7) geben 60 Bilder pro Sekunde als typischen Wert an, der die Forderung nach Echtzeit erfüllt.

Zusammenfassend sind die Hauptmerkmale von Virtual Reality:

- eine in Echtzeit computergenerierte virtuelle Welt
- die vom Betrachter abhängige Präsentation dieser Welt
- das Ansprechen von multimodalen Reizen
- Möglichkeit zur Interaktion mit der virtuellen Welt
- durch das Zusammenspiel dieser Merkmale das Ermöglichen eines immersiven Erlebens der virtuellen Welt und des Gefühls sich innerhalb dieser Welt zu befinden, von ihr umgeben zu sein

In dieser Arbeit wird, wenn von Virtual Reality oder dem Einsatz von Virtual Reality gesprochen wird, der Einsatz eines VR-Systems bezeichnet, welches die zuvor genannten Merkmale aufweist.

2.3.2 Einschränkungen von Virtual Reality Systemen

Neben den Vorteilen, die Virtual Reality bietet, wie die Immersion in die virtuelle Umgebung und das virtuelle Geschehen, hat sie auch Einschränkungen. Zwei wichtige Faktoren, die beim Einsatz von Virtual Reality zu beachten sind, sind die Hardware und teilweise auch die Platzanforderungen, die durch die VR-Systeme gestellt werden. Für die Verwendung von VR-Anwendungen ist ein VR-System notwendig, das aus verschiedenen Komponenten bestehen kann. Für einfache VR-Anwendungen am Smartphone ist nur ein Headset wie Daydream View von Google (Google LLC, o.D.-a) notwendig. Diese Headsets registrieren die Kopfdrehung und -neigung, aber nicht die Position im Raum (Janssen, 2019). Für komplexere Anwendungen, die beispielsweise das Tracking des Nutzers und der Controller im Raum ermöglichen und dadurch die Fortbewegung in der virtuellen Umgebung durch die Fortbewegung im realen Raum simulieren, war bisher ein VR-System mit mindestens zwei externen Sensoren notwendig, wie das der HTC Vive oder Oculus Rift (Janssen, 2019). Ein solches System benötigt jedoch die Verbindung zu einem Computer, der die nötige Leistung erbringt (siehe Tabelle 2-4, S. 18), und für möglichst viel Bewegungsfreiheit eine entsprechend große freie Fläche. Dadurch können für komplette VR-Systeme große Hardwarekosten entstehen. Das HTC Vive Pro System inklusive Controllern und Tracking-Hardware kostet aktuell auf der Herstellerseite 1.399 Euro (HTC Corporation, 2019a), während der Preis des Nachfolgers der Oculus Rift, der

Oculus Rift S, vom Hersteller mit 449 Euro angegeben wird (Facebook Technologies LLC, o.D.-b). Hinzu kommen die Kosten für einen entsprechend ausgerüsteten Computer.

Prozessor	Intel® Core™ i5-4590 oder AMD FX™ 8350, vergleichbar oder besser
Grafikkarte	NVIDIA® GeForce® GTX 970 oder AMD Radeon™ R9 290, vergleichbar oder besser
Arbeitsspeicher	4 GB RAM oder mehr
Betriebssystem	Windows® 7, Windows® 8.1 oder aktueller, Windows® 10

Tabelle 2-4: PC Mindestanforderungen für eine HTC Vive Pro (Inhalt nach (HTC Corporation, 2019a))

Oculus hat mit der Oculus Quest ein Head-Mounted Display vorgestellt, welches keine Anbindung an einen PC und auch kein Smartphone benötigt (Facebook Technologies LLC, o.D.-a). Dabei nutzt die Oculus Quest sogenanntes Insight Tracking, wodurch ein Tracking im Raum ohne externe Sensoren ermöglicht wird (Facebook Technologies LLC, o.D.-a). Auf diesem System können jedoch nicht alle Anwendungen verwendet werden, die auf einem System mit entsprechender PC-Anbindung verfügbar sind, unter anderem da die Oculus Quest für manche Programme nicht die nötige Leistung zur Verfügung stellt (Robertson, 2019).

Des Weiteren ergeben sich Einschränkungen in der Wahrnehmung der virtuellen Welt im Vergleich zur Realität, etwa durch ein eingeschränktes Sichtfeld bei den Head-Mounted Displays. Das binokulare Gesichtsfeld eines Menschen beträgt horizontal ca. 180 Grad (Broll, 2013, S. 278), für die HTC Vive und Vive Pro gibt der Hersteller ein Sichtfeld von 110 Grad (HTC Corporation, 2019b; HTC Corporation, 2019a), für die PSVR ein Sichtfeld von 100 Grad an (Sony Interactive Entertainment Europe Limited, 2019). Dabei ist auch der Sichtbereich, der als scharf wahrgenommen wird und in dem beispielsweise Schrift gut erkannt werden kann, bei der Nutzung von Head-Mounted Displays bisher kleiner als in der Realität.

Eine weitere Einschränkung bietet das teilweise mit der Zeit unangenehme Tragen der Head-Mounted Displays und das Auftreten von Symptomen der Reise- bzw. Bewegungs-krankheit bei manchen Nutzern und Anwendungen.

2.3.3 Einsatzgebiete von Virtual Reality Systemen

Virtual Reality, meist unter dem Einsatz von HMDs, wird heute in verschiedenen Bereichen eingesetzt. Durch die Einführung des HMDs Playstation VR für die Spielekonsole Playstation 4 von Sony Interactive Entertainment und den Vertrieb von VR-Anwendungen über die Online-Plattform Steam, die hauptsächlich Spiele umfasst (Valve Corporation, 2019c), ist Virtual Reality inzwischen im Gaming Bereich vertreten.

Des Weiteren wird Virtual Reality in verschiedenen Bereichen als Lernmedium eingesetzt. In der Industrie werden VR-Anwendungen genutzt, um Mitarbeiter anzulernen, da dadurch beispielsweise der Ausfall von Maschinen während der Lerneinheiten vermieden werden kann (Zobel, Werning, Metzger, & Thomas, 2018, S. 135). Dede, Salzman und Loftin (1996, S. 252) merkten in den Ergebnissen ihrer Untersuchung in Bezug auf Virtual Reality zum Lernen von komplexen und abstrakten wissenschaftlichen Konzepten an, dass Studenten virtuelle Welten als attraktive Lernumgebungen empfinden. Inzwischen existieren VR-Anwendungen, die beispielsweise im Bereich des Ingenieurwesens als Lernmedium eingesetzt werden können (Abulrub, Attridge, & Williams, 2011). In der Astronomie stellten Mintz, Litvak und Yair (2001) eine 3D-Visualisierung des Sonnensystems in VR als Lernumgebung vor, die genutzt wird, um es für Lernende begreifbarer zu machen. Als einer der Vorteile von VR wird hierbei die einfache Änderung der eigenen Perspektive auf das Visualisierte genannt (Mintz et al., 2001). Witmer und Singer (1998, S. 239) äußern in Zusammenhang mit ihrer Untersuchung von Immersion und Präsenz in VR-Anwendungen die Vermutung, dass immersive Umgebungen sich besser als Trainings-/Lernanwendungen eignen, als Standard Computer-Programme. Auch Psotka (1995, S. 409) äußert sich ähnlich, indem er angibt, dass die herausragende Eigenschaft, die VR im Vergleich zu anderen Lernanwendungen bietet, die Möglichkeit zur Generierung von Immersion ist. Chu, Humer und Eckhardt (2019) befassten sich in einer Studie mit der Implementierung und Evaluierung einer VR-Anwendung zur Vermittlung der speziellen Relativitätstheorie. Als Ergebnisse ihrer Studie geben sie an, dass die Visualisierung in VR zu einer hohen Immersion der Nutzer und einem erhöhten Wissen über die spezielle Relativitätstheorie geführt hat (Chu, Humer, & Eckhardt, 2019). Dadurch könnte VR bei der Vermittlung von komplexen Inhalten einen Vorteil bieten.

3 Konzept

In diesem Kapitel wird das Konzept zur Visualisierung eines CNNs in Virtual Reality vorgestellt, das im Rahmen dieser Arbeit entwickelt wurde. In Abschnitt 3.1 wird das Nutzerszenario für die Anwendung beschrieben. Darauf folgen in Abschnitt 3.2 die grundlegenden Gedanken und Fragestellungen, aus denen das Konzept entstanden ist. Anschließend wird in Abschnitt 3.3 das Konzept für die Visualisierung anhand des Visual Information Seeking Mantras von Ben Shneiderman beschrieben und zum Abschluss des Kapitels in Abschnitt 3.4 auf die gewählte Interaktionstechnik eingegangen.

3.1 Nutzerszenario

Die Anwendung zur Visualisierung eines CNNs richtet sich an technisch versierte Nutzer, die beispielsweise im Bereich der Informatik tätig sind und in den Bereich der Künstlichen Neuronalen Netze einsteigen wollen. Denkbar wäre beispielsweise, dass im Rahmen eines neuen Projektes Künstliche Neuronale Netze eingesetzt werden sollen und der Nutzer daher mehr darüber erfahren möchte. Der Nutzer ist dabei kein Experte auf dem Gebiet des Maschinellen Lernens und der Künstlichen Neuronalen Netze, hat aber eventuell schon erstes Vorwissen in den Bereichen. Die Anwendung soll dabei speziell den Einstieg in die Funktionsweise von Convolutional Neural Networks begleiten. Sie soll dabei jedoch nicht als alleiniges Lernmedium, sondern als additives Lernverfahren dienen, um mit einem neuen Ansatz die Funktionsweise von CNNs transparenter zu machen.

3.2 Grundlagen

Das Konzept für die Visualisierung basiert auf der Annahme, dass sich beim Einsatz von Virtual Reality bestimmte Vorteile ergeben, die bei einer 2D- oder auch 3D-Visualisierung auf einem normalen Bildschirm nicht vorhanden sind. Beim Einsatz von VR-Anwendungen mit einem Head-Mounted-Display ist ein Effekt, der im Vergleich zu herkömmlichen Anwendungen am PC-Bildschirm verstärkt wird, die Immersion. In einer VR-Anwendung hat der Nutzer im Optimalfall das Gefühl, sich tatsächlich in der virtuellen Welt zu befinden und empfindet sie als real. Durch die Ausblendung der realen Umwelt könnte eine höhere Konzentration auf die virtuellen Inhalte entstehen und der Nutzer könnte sich dadurch intensiver mit ihnen auseinandersetzen. Durch diese Immersion ergeben sich neue

Möglichkeiten für die Nutzung des dreidimensionalen Raumes. Das Konzept probiert einen reduzierten grafischen Ansatz unter Ausnutzung der Effekte der Immersion für den Betrachter. Für die Visualisierung steht eine größere Fläche zur Verfügung, durch die der Nutzer auf natürliche Art und Weise navigieren kann. Durch Kopfbewegungen und die Bewegung seines Körpers kann er seine Perspektive ändern und andere Bereiche der virtuellen Welt betrachten, als würde er sich in einem realen Raum befinden.

Dieses Visualisierungskonzept konzentriert sich darauf, diesen neuen verfügbaren Raum durch eine übersichtliche Darstellung relevanter Informationen möglichst sinnvoll zu nutzen. Die Tiefe im Raum bietet die Möglichkeit, Informationen auf verschiedenen Ebenen hintereinander zu präsentieren. Der gesamte Raum, der den Nutzer umgibt, kann zum Platzieren von Informationen genutzt werden, wodurch eine größere Anzahl an Informationen gleichzeitig und trotzdem übersichtlich betrachtet werden könnte. Auch schon vorhandene 2D-Informationen könnten dem Nutzer in einer anderen, und möglicherweise informativeren, Sichtweise dargeboten werden.

Durch die Visualisierung sollen der Aufbau und die Funktionsweise eines CNNs dargestellt werden. Da die Convolutional- und Pooling-Layer die Besonderheit eines CNN darstellen, liegt der Fokus des Konzepts auf ihnen. Die Idee ist es, die Visualisierung an ein reales CNN zur Bilderkennung anzubinden und die darzustellenden Daten zur Laufzeit von dem Modell abzurufen. Bei dem Modell soll es sich um ein bereits trainiertes CNN handeln, im Gegensatz zu den Visualisierungen, die den Lernprozess darstellen, wie TensorBoard (siehe Abschnitt 2.2.2) und sich vor allem an Entwickler richten. In der Visualisierung soll der Nutzer die Möglichkeit haben, verschiedene Inputbilder auszuwählen und sie vom CNN klassifizieren zu lassen. Für die verschiedenen Layer des CNNs soll er sich dann die Zwischenergebnisse für das jeweilige Inputbild anzeigen lassen können sowie den Output des Modells. Dadurch soll der Nutzer einen Einblick in das Verhalten des CNNs bei verschiedenen Inputbildern und somit in die Funktionsweise des CNNs bekommen, wie es sich auch Harley bei seiner Visualisierung zum Ziel gesetzt hat (siehe Abschnitt 2.2.4).

Eine wichtige Frage während der Konzeptionsphase war, wie tief die Visualisierung in die Details eines CNNs gehen soll, ob beispielsweise die einzelnen Neuronen visualisiert werden und ihre einzelnen Werte abrufbar sein sollen. Shneiderman (Shneiderman, o.D.)

empfiehlt in Hinblick auf die Gestaltung von Benutzeroberflächen eine Reduzierung des Datenvolumens, das dem menschlichen Gehirn zur Informationsverarbeitung angeboten wird. Er begründet dies damit, dass Menschen nur eine limitierte Menge an Informationen im Kurzzeitgedächtnis zwischenspeichern können (Shneiderman, o.D.). In Hinblick auf die Zielsetzung der Anwendung als Lernumgebung zu dienen, wurde schließlich eine Reduzierung der Informationen und eine Konzentration auf die wichtigsten, grundlegenden Aspekte eines CNNs beschlossen. Es wurde sich bewusst dagegen entschieden, die einzelnen Neuronen des CNNs und ihre Verbindungen anzuzeigen. Stattdessen sollen die Informationen der einzelnen Layer für jeden Layer übersichtlich zusammengefasst werden, wie es auch bei der ConvNetJS Demo von Karpathy der Fall ist (siehe Abschnitt 2.2.3).

Der Fokus der Visualisierung soll auf den für das Verständnis des CNNs relevanten Informationen liegen. Bei CNNs spielen die gelernten Filter der Convolutional-Layer und die entstehenden Feature-Maps eine entscheidende Rolle. Dementsprechend konzentriert sich die auf diese, indem die Filter und Feature-Maps visuell als Bilder dargestellt werden. Des Weiteren soll der Aufbau des CNNs, also die Abfolge der verschiedenen Layer, und die Größendimensionen der Layer dargestellt werden. Für den Nutzer soll es ersichtlich werden, wie sich die Inputdaten im Laufe der Prozesse des CNNs verändern. Bei der Frage welche Informationen dargestellt werden, wurde sich vor allem an Karpathys Visualisierung orientiert, da sie als sehr übersichtlich empfunden wurde. So werden Parameterdetails, wie Filtergröße und *Stride* zu den entsprechenden Layern angezeigt, Feature-Maps visualisiert und die minimale und maximale Aktivierung der Feature-Maps dargestellt.

3.3 Aufbau der Visualisierung

Das Konzept orientiert sich an den Grundprinzipien für die Visualisierung von Informationen von Ben Shneiderman (1996). Shneiderman fasst sein sogenanntes „Visual Information Seeking Mantra“ (1996) folgendermaßen zusammen:

“Overview first, zoom and filter, then details-on-demand.” (Shneiderman, 1996, S. 337)

Er erläutert diese Oberbegriffe seines Mantras wie folgt:

“Overview: Gain an overview of the entire collection.

Zoom: Zoom in on items of interest

Filter: filter out uninteresting items.

Details-on-demand: Select an item or group and get details when needed.” (Shneiderman, The eyes have it: A task by data type taxonomy for information visualizations, 1996, S. 337)

Nach diesen Prinzipien wurde das Visualisierungskonzept erstellt und wird anhand der Oberbegriffe im Folgenden erläutert. Abbildung 3-1 bietet einen Überblick über den Aufbau der Visualisierung.

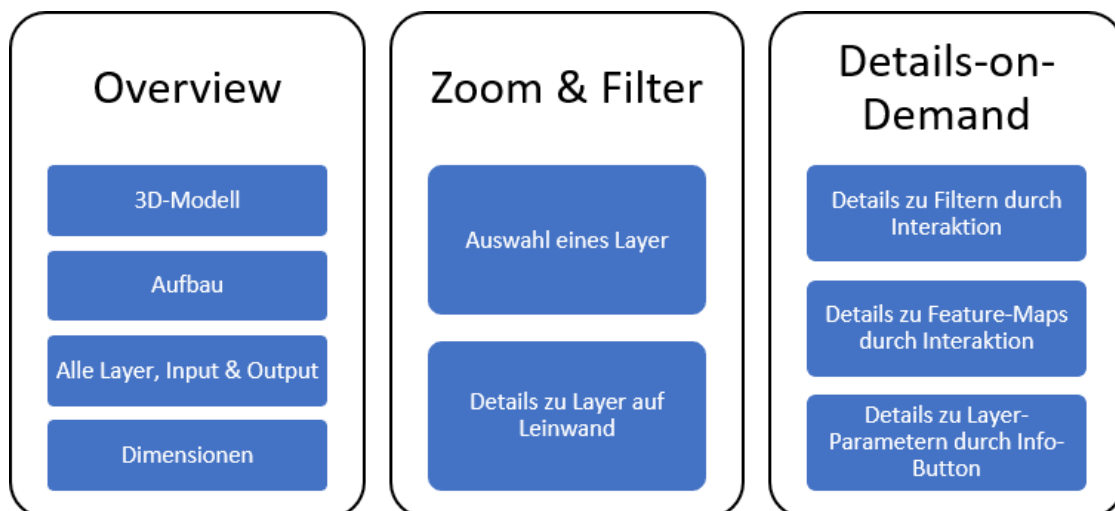


Abbildung 3-1: Übersicht über den Aufbau der Visualisierung

Overview:

Um einen Überblick über das präsentierte CNN-Modell zu schaffen wurde ein 3D-Modell gewählt, welches den Aufbau des CNNs anhand der einzelnen Layer aufzeigt (siehe Abbildung 3-2). Die Darstellung orientiert sich an der 2D-Darstellung von LeNet-5 von LeCun et al. (1998) (siehe Abschnitt 2.2.1). Die einzelnen Feature-Maps der Convolution- und Pooling-Layer werden für eine bessere Skalierbarkeit in Boxen zusammengefasst. Jeder dieser Layer wird somit als eine Box dargestellt, deren Größendimensionen sich an den Dimensionen des Layer orientieren. Die Dimensionen der Convolutional- und Pooling-Layer beziehen sich dabei auf die Größe des Outputs, der durch die Operation auf dem Layer entsteht. Bei einem Convolutional-Layer wird mit der Beschriftung 24x24x32

beispielsweise angegeben, dass durch die Convolution-Operation auf diesem Layer, 32 Feature-Maps mit den Maßen 24x24 entstehen, so wie es auch in der ConvNetJS Demo von Karpathy gemacht wird (siehe Abschnitt 2.2.3). Der Input- und der Output-Layer sind davon ausgenommen, sie stellen direkt die Dimensionen des Modell-Inputs und -Outputs dar. Bei dem Fully-Connected-Layer gibt die Beschriftung die Anzahl der Neuronen des Layers an. Durch diese Darstellung soll der Nutzer auf einen Blick den Aufbau des Modells und die Dimensionsänderung der Daten innerhalb des CNNs erfassen können.

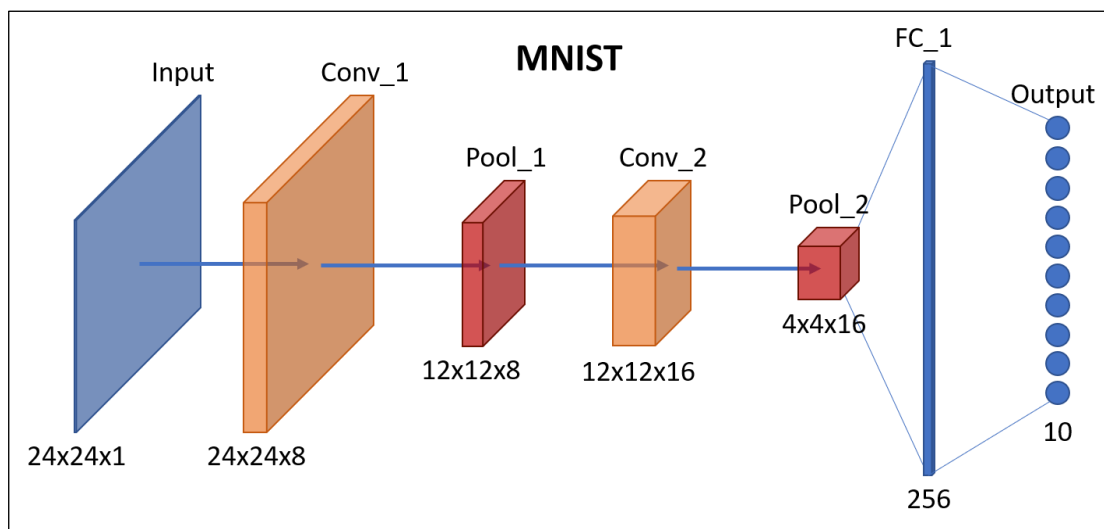


Abbildung 3-2: Erste Konzeptdarstellung des 3D-Modells des CNNs

Die verschiedenen Arten von Layern haben dabei jeweils eine eigene Farbe, um eine schnellere Orientierung zu ermöglichen, wie es auch bei der ConvNetJS Demo der Fall ist. So besitzen die Boxen der Convolutional-Layer eine orange Färbung, die Pooling-Layer eine rote und die Fully-Connected-Layer eine blaue Färbung. Der Input-Layer ist in der Prototyp-Anwendung durch eine Darstellung des Inputbildes abgebildet. Die Beschriftungen am Modell geben die genauen Dimensionen und die Bezeichnungen der Layer an.

Zoom and Filter:

Über das 3D-Modell kann gesteuert werden, zu welchem Layer Details angezeigt werden sollen. Bei der Auswahl eines Layers erscheint hinter dem 3D-Modell eine Leinwand, die den Input, die Operation und den Output des Layers darstellt. In Abbildung 3-3 (S. 25) ist der erste Entwurf für das Konzept dargestellt. Die genaue Darstellung ist dabei abhängig von der Art des Layers. Die Hintergrundfarbe der Leinwände entspricht der Farbe des

Layers im 3D-Modell, um die Zuordnung und den Überblick darüber zu welchem Layer die Details gehören, zu erleichtern.

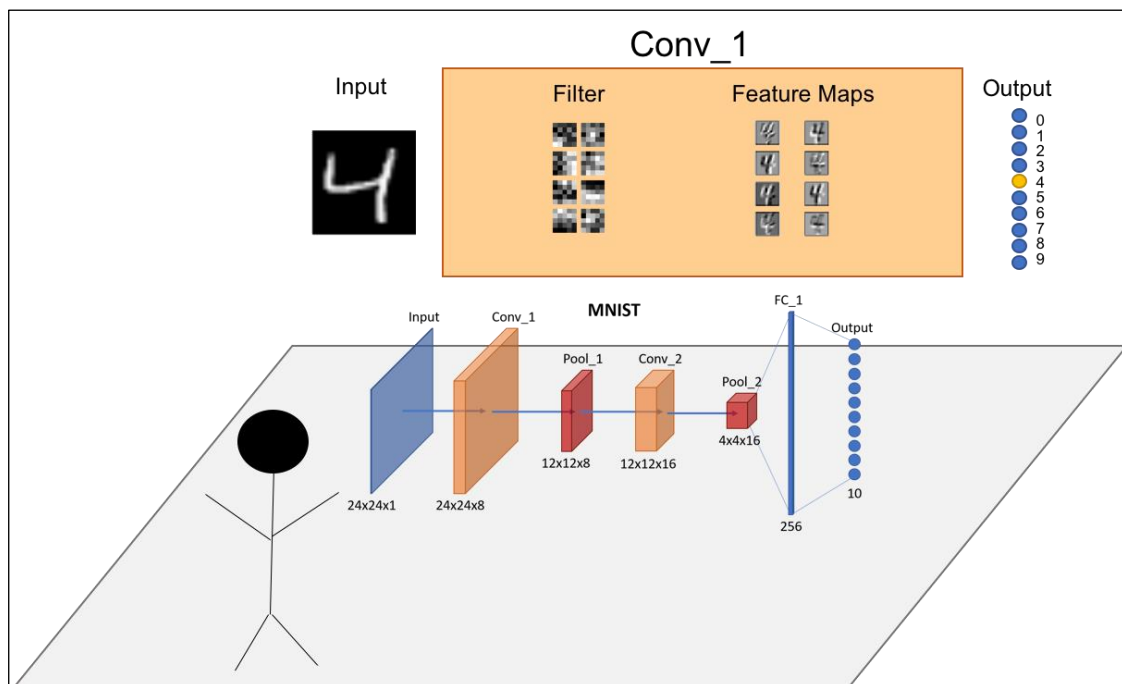


Abbildung 3-3: Konzeptbild zur Darstellung eines Convolutional-Layers

Der Input-Layer zeigt das derzeit ausgewählte Inputbild und dessen Dimensionen. Über einen Button kann ein anderes Inputbild ausgewählt werden. Als Input des Convolutional-Layers wird entweder das gewählte Inputbild angezeigt, wenn es der erste Layer nach dem Input ist, oder die Output-Feature-Maps des vorherigen Convolutional- oder Pooling-Layers. Als Operation werden die gelernten Filter des Layers angezeigt (für eine Beschreibung der Visualisierung der Filter und Feature-Maps siehe Abschnitt 4.3, S. 34). Als Output werden die Feature-Maps angezeigt, die auf dem Layer entstehen. Für eine Auflistung der dargestellten Inhalte aller Layer siehe Tabelle 3-1 (S. 26).

Art des Layers	Input-Anzeige	Operation-Anzeige	Output-Anzeige
Input	Derzeit ausgewähltes Inputbild und dessen Dimensionen	-	-
Convolutional	Das ausgewählte Inputbild oder die Output-Feature-Maps des vorherigen Convolutional- oder Pooling-Layers	Die gelernten Filter des Layers	Die auf dem Layer entstandenen Feature-Maps
Pooling	Output-Feature-Maps des vorherigen Convolutional- oder Pooling-Layers	Erklärgrafik zum Max-Pooling Verfahren	Die verkleinerten Feature-Maps
Fully-Connected	Output-Feature-Maps des vorherigen Convolutional- oder Pooling-Layers	Grafik mit beispielhafter Darstellung der Neuronen und ihrer Verbindungen	Output-Neuronen mit Markierung der klassifizierten Ziffer mit Angabe der Wahrscheinlichkeit
Output	-	-	Output-Neuronen mit Markierung der klassifizierten Ziffer mit Angabe der Wahrscheinlichkeit

Tabelle 3-1: Übersicht über die Layer-Arten und die dargestellten Inhalte

Details-on-demand:

Die Filter und Feature-Maps auf den Detailleinwänden können ausgewählt werden, um eine Detailanzeige zu ihnen einzublenden. Diese Anzeige erscheint rechts oder links neben den großen Leinwänden. Auf der Detailanzeige wird die jeweilige Visualisierung vergrößert dargestellt und eine Farbskala, die anzeigt, welche Farbe in der Visualisierung welchem Zahlenwert entspricht. Der Minimal- und Maximalwert der Farbskala entspricht dabei der minimalen und maximalen Aktivierung auf dem Layer. Zu den Filtern erscheint außerdem eine Matrix mit den entsprechenden Gewichtswerten. Auf dem Convolutional-

Layer werden bei der Detailansicht jeweils der Filter und die daraus entstehende Feature-Map übereinander angezeigt, um einen direkten Vergleich zu ermöglichen.

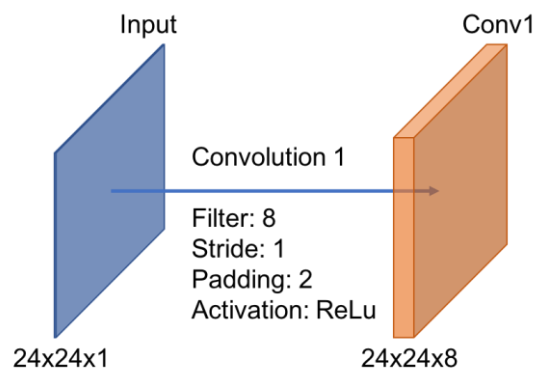


Abbildung 3-4: Konzeptbild für kleines 3D-Modell mit Parameterdetails

Am 3D-Modell zum Aufbau des CNNs befinden sich zwischen der Input- und Convolutional-Layer und zwischen der Convolutional- und Pooling-Layer kleine Info-Buttons. Mit diesen können Details zu den Parametern der Operation angezeigt werden, die zwischen den Layern ausgeführt wird. Für diese Detailanzeigen wird ein kleines 3D-Modell verwendet, dass neben dem großen Modell erscheint (siehe Abbildung 3-4). Das kleine Modell zeigt die Input- und die Output-Layer der Operation mit Beschriftungen zu ihren Dimensionen und Bezeichnungen an, zusätzlich werden zwischen den Layern die Parameter der Operation eingeblendet. Bei einer Convolution-Operation werden beispielsweise die Anzahl der Filter, die Werte für *Stride* und *Padding* und die Aktivierungsfunktion angezeigt. Somit bietet die Detailanzeige eine Übersicht über die Dimensionsänderung vom Input zum Output der Operation und die eingestellten Parameter. Alle Detailanzeigen können einzeln ausgeblendet werden.

3.4 Interaktionstechnik

Als Interaktionstechnik in der virtuellen Umgebung wurde ein Raycast-Pointer gewählt. Dabei verwendet der Nutzer eine Art Laserstrahl, der mit dem VR-Controller verbunden ist, um mit User Interface Elementen in der virtuellen Umgebung zu interagieren. Der Pointer liefert den Vorteil, dass in einem gewissen Umfang, unabhängig davon, wie weit entfernt man sich von interaktiven Elementen befindet, immer mit ihnen interagiert werden kann. Das bedeutet, dass der Nutzer sich nicht zu einem bestimmten Element hinbewegen muss, um damit zu interagieren, was ihm Bewegungs- und Positionierungsspielraum in der

virtuellen Umgebung verschafft. Die Interaktion soll dabei möglichst einfach gestaltet werden, um auch Nutzern ohne VR-Erfahrung eine intuitive Bedienung der Anwendung zu ermöglichen. Daher soll der Pointer immer aktiv sein und nicht durch eine zusätzliche Interaktion aktiviert werden müssen. User Interface Elemente mit denen interagiert werden kann, sollen dies durch eine Skalierungsanimation beim Kontakt mit dem Pointer signalisieren. Zeigt der Pointer auf ein interaktives Element soll damit mit nur einem Button des Controllers interagiert werden können. Die Farbe des Pointers soll so gewählt werden, dass sie sich von der virtuellen Umgebung entsprechend abhebt, um für den Nutzer gut sichtbar zu sein.

4 Umsetzung

In diesem Kapitel wird die Umsetzung des in Kapitel 3 beschriebenen Konzepts als Prototyp erläutert.

4.1 Software und Hardware

Im Folgenden wird die Soft- und Hardware beschrieben, die für die Entwicklung der VR-Anwendung verwendet wurde.

4.1.1 Entwicklungs-PC

Der PC, der für die Entwicklung der VR-Anwendung genutzt wurde, hatte folgende Spezifikationen:

- Betriebssystem: Windows 10 Home, 64-Bit
- Prozessor: Intel® Core™ i7-4770K CPU @ 3.50GHz
- RAM: 16,0 GB
- Grafikkarte: NVIDIA GeForce RTX 2070, 8,0 GB

4.1.2 HTC Vive / Vive Pro

Die VR-Anwendung wurde während der Entwicklung mit einer HTC Vive und einer HTC Vive Pro getestet. Die Evaluierung der Anwendung wurde mit einer HTC Vive Pro durchgeführt (siehe Abschnitt 5.2). Bei der HTC Vive und der HTC Vive Pro handelt es sich um Head-Mounted Displays vom Hersteller HTC in Zusammenarbeit mit Valve, die in Kombination mit einem Tracking-System und Controllern für die Nutzung von VR-Anwendungen entwickelt wurden (HTC Corporation, 2019a, 2019b). Eine Zusammenstellung der wichtigsten technischen Daten der beiden HMDs können der Tabelle 4-1 (S. 30) entnommen werden.

	HTC Vive	HTC Vive Pro
Bildschirm	Dual AMOLED 3,6'' diagonal	Dual AMOLED 3.5'' diagonal
Auflösung	1080 x 1200 Pixel pro Auge (2160 x 1200 Pixel zusammen)	1440 x 1600 Pixel pro Auge (2880 x 1600 Pixel zusammen)
Bildwiederholrate	90 Hz	90 Hz
Sichtfeld	110 Grad	110 Grad
Sensoren	SteamVR Tracking, G-Sensor, Gyroskop, Nähesensor	SteamVR Tracking, G-Sensor, Gy- roskop, Entfernungssensor, IPD Sensor

Tabelle 4-1: Technische Daten der HTC Vive und Vive Pro (Inhalt nach (HTC Corporation, 2019a, 2019b))

Die HTC Vive / Vive Pro wurden als Hardware gewählt, da sie durch das DLR und die Hochschule Düsseldorf zur Verfügung standen und durch vorherige VR-Projekte bereits Erfahrung im Umgang damit vorhanden war.

4.1.3 Unity

Das Visualisierungskonzept wurde mit der Spiele-Engine Unity von Unity Technologies als VR-Anwendung umgesetzt. Die Unity Engine wurde verwendet, da sie beim DLR für die Entwicklung genutzt wird und es daher eine Anforderung an die Anwendung ist, dass sie mit dieser Entwicklungsumgebung weiterentwickelt werden kann.

Unity ist eine Echtzeit-3D-Entwicklungsplattform, mit der Anwendungen für über 25 verschiedene Plattformen entwickelt werden können, wie Windows- und OS X Desktop-Systeme, aber auch für mobile Plattformen und einige Spielekonsolen (Unity Technologies, 2019b). Die Engine kann durch Assets wie beispielsweise Plugins und 3D-Modelle erweitert werden, die über den zugehörigen Asset-Store heruntergeladen werden können (Unity Technologies, 2019b). Der Unity-Editor, der für die Entwicklung mit der Unity Engine genutzt wird, bietet einige Tools, welche die Erstellung von 2D- und 3D-Anwendungen unterstützen. Der Editor beinhaltet dabei sowohl Tools für die Erstellung von Spielwelten und Benutzeroberflächen wie das integrierte UI-System, als auch Entwickler-Tools zur Implementierung der Spiellogik (Unity Technologies, 2019b). Für die Entwicklung einer Anwendung wird in Unity ein Projekt erstellt, dem im Entwicklungsverlauf alle Bestandteile der Anwendung hinzugefügt werden. Der Hauptbestandteil eines Unity-Projektes sind

die sogenannten *Scenes* (im Folgenden Szenen), welche die virtuelle Umgebung und die Benutzeroberfläche der Anwendung enthalten (Unity Technologies, 2019c). Auch die Skripte, welche die Logik der Anwendung implementieren, sind Bestandteil der Szenen und können durch entsprechende Verknüpfungen auf die Elemente der virtuellen Umgebung und der Benutzeroberfläche zugreifen. Jede Szene repräsentiert dabei einen Level des Spiels bzw. der Anwendung (Unity Technologies, 2019c), die im Unity Editor miteinander verknüpft werden können.

Für die Entwicklung wurde Unity in der Version 2018.3.5 genutzt, die unter (Unity Technologies, 2019a) verfügbar ist.

4.1.4 SteamVR und SteamVR Unity Plugin

SteamVR ist ein kostenloses Softwareprodukt der Valve Corporation, welches die Nutzung von VR-Anwendungen auf kompatibler Hardware ermöglicht (Valve Corporation, 2019a). Unterstützt werden dabei Head-Mounted Displays, wie die HTC Vive, die Oculus Rift und das Windows Mixed Reality Headset (Valve Corporation, 2019a). SteamVR bietet, neben der Möglichkeit VR-Anwendungen auszuführen, auch eine Übersicht über den Status der VR-Hardware, wie das HMD und die Controller, und bietet die Möglichkeit den Bewegungsbereich während der Nutzung von VR-Anwendungen festzulegen (Valve Corporation, 2019a). Dieser Bewegungsbereich kann dann für alle VR-Anwendungen genutzt werden, die mit SteamVR gestartet werden. Kommt der Nutzer während einer VR-Anwendung an den Rand dieses Bereiches, wird dies durch die Einblendung eines Gitternetzes in der virtuellen Umgebung signalisiert. Falls nicht genug Platz für einen solchen Bewegungsbereich vorhanden ist, kann auch eine sitzende oder stehende Position für die Nutzung von VR-Anwendungen eingerichtet werden.

Die Valve Corporation stellt außerdem das SteamVR Unity Plugin für Entwickler zur Verfügung, mit dem VR-Anwendungen für VR-Systeme wie die HTC Vive mit Unity entwickelt werden können (Valve Corporation, o.D.). Durch das Plugin wird eine Verbindung zu SteamVR geschaffen und darüber zu einem unterstützten VR-System (Valve Corporation, o.D.). Dadurch kann in Unity auf alle für die Entwicklung benötigten Elemente des VR-Systems zugegriffen werden. Somit kann beispielsweise der Input der Controller verarbeitet und 3D-Modelle der Controller in die virtuelle Umgebung eingefügt werden (Valve Corporation, o.D.). Unter Verwendung von SteamVR und dem SteamVR

Plugin für Unity können außerdem sich in der Entwicklung befindende VR-Anwendungen mit Unity ausgeführt und getestet werden.

Das Plugin kann kostenlos aus dem Unity Asset-Store heruntergeladen und in einem Unity Projekt installiert werden (Valve Corporation, 2019b).

Die VR-Anwendung, die im Rahmen dieser Arbeit entwickelt wurde, wurde zuletzt mit dem SteamVR Unity Plugin in Version 2.3.2 getestet.

4.1.5 TensorFlow & Keras

TensorFlow ist ein Open-Source Software-Framework, das als Schnittstelle zur Beschreibung und Implementierung maschineller Lernalgorithmen eingesetzt werden kann (Hope, Resheff, & Lieder, 2018, S. 6). Die Kernkomponente des Frameworks ist in C++ geschrieben und besitzt Schnittstellen zu Hochsprachen, wobei die Schnittstelle zur Hochsprache Python am weitesten entwickelt ist (Hope et al., 2018, S. 6). Ein Fokus des Frameworks liegt auf der Portierbarkeit, um die Ausführung von Berechnungsgraphen in vielen unterschiedlichen Umgebungen und auf vielen verschiedenen Hardwareplattformen zu gewährleisten (Hope et al., 2018, S. 6).

Keras ist eine high-level Neural Networks API, die in Python geschrieben wurde und unter anderem über TensorFlow laufen kann (Keras-Team, o.D.-b). Die API ermöglicht einen einfacheren Zugriff auf TensorFlow, beispielsweise erlaubt Keras es, Machine-Learning-Modelle mit wenigen Codezeilen zu erstellen und anzulernen (Hope et al., 2018, S. 7).

4.2 Das CNN-Modell

Die Daten, auf denen die Visualisierung basiert, wurden von einem CNN-Modell übernommen, welches mit dem Framework Keras in einem Python-Skript erstellt wurde. Der Code wurde größtenteils von (Keras-Team, 2018) übernommen und leicht abgewandelt, sodass das verwendete Modell nur einen Convolutional-Layer besitzt statt zwei und nach der Trainings- und Testphase im h5-Format gespeichert wird. Der Quellcode zur Erstellung des Modells kann Abbildung 4-1 (S. 33) entnommen werden.

```

model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3),
                activation='relu',
                input_shape=input_shape))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))

```

Abbildung 4-1: Quellcode zur Generierung des CNN-Modells mit Keras

Eine Übersicht über den Aufbau des Modells liefert die Keras-Funktion `model.summary()`. Die Ausgabe dieser Funktion für das verwendete CNN-Modell ist in Abbildung 4-2 dargestellt.

Das Modell wurde mit den Daten der „MNIST database for handwritten digits“ (Keras-Team, o.D.-a) trainiert, um Graustufenbilder handschriftlicher Ziffern von 0-9 klassifizieren zu können. Der MNIST Datensatz besteht aus sechzigtausend Bildern in der Größe 28x28, mit einem Testdatensatz von zehntausend Bildern (Keras-Team, o.D.-a).

Es wurde dieser einfachere Aufbau für den Prototyp gewählt, um zunächst das Visualisierungskonzept für die unterschiedlichen Layer zu testen.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d_1 (MaxPooling2D)	(None, 13, 13, 32)	0
dropout_1 (Dropout)	(None, 13, 13, 32)	0
flatten_1 (Flatten)	(None, 5408)	0
dense_1 (Dense)	(None, 128)	692352
dropout_2 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 10)	1290
Total params: 693,962		
Trainable params: 693,962		
Non-trainable params: 0		

Abbildung 4-2: Aufbau des verwendeten CNNs, Ausgabe der Keras-Funktion `model.summary()`

4.3 Visualisierung der Filter und Feature-Maps

Um die Filter und Feature-Maps des CNNs zu visualisieren wurde ein Vorgehen gewählt, das von Michael A. Irvine in einem Blogbeitrag vorgestellt wurde (Irvine, 2017). Der Code, der zur Generierung der Filter- und Feature-Map-Bilder genutzt wurde, basiert zum Großteil auf seinem Code. Irvines Vorgehen wurde außerdem mit dem Verfahren abgeglichen, welches der Entwickler der Keras-Bibliothek, François Chollet, in seinem Buch *Deep Learning mit Python und Keras* beschreibt (2018, S. 210-217). Dabei werden die Feature-Maps jedes Convolutional- und Pooling-Layers mithilfe der Python-Bibliothek *matplotlib* als 2D-Bilder dargestellt (Chollet, 2018).

Um die Aktivierungen eines bestimmten Layer als Ausgabe zu bekommen, statt der *Predictions* für die zehn Klassen, wurden zwei neue Keras-Modelle erstellt, die dem ursprünglichen Modell gleichen bis auf die Tatsache, dass sie mit dem Layer enden, dessen Aktivierungen visualisiert werden sollen. Die gelernten Gewichte für den Convolutional-Layer werden übernommen. Eines der Modelle endet mit dem Convolutional-Layer und gibt dessen Aktivierungen aus, und das andere endet mit dem Pooling-Layer und gibt wiederum dessen Aktivierungen aus. Wenn man das Modell eine *Prediction* ausführen lässt, erhält man als Output einen 4D-Tensor, welcher die Aktivierungswerte der Feature-Maps enthält, geordnet nach Batch, Reihe, Spalte und Filter/Kanal. Dabei gleicht die Anzahl der Kanäle der Anzahl der Filter auf dem Convolutional-Layer. Für jeden der Filter kann also eine Feature-Map visualisiert werden. Für jeden Kanal ein eigenes 2D-Bild zu erstellen, ist laut Chollet angebracht, da sie „relativ unabhängige Merkmale“ (Chollet, 2018, S. 211) codieren.

Mit Hilfe der *matplotlib* Funktion *imshow* kann den Aktivierungswerten für eine Feature-Map abhängig von ihrem Wert ein Farbwert einer *colormap* zugewiesen werden und die Feature-Map dann als Bild gespeichert werden. Für die Anwendung wurden die Bilder im PNG-Format gespeichert. Durch diese Visualisierung werden die Aktivierungswerte visuell sichtbar und es wird beispielsweise ersichtlich, auf welche Bildbereiche der zugehörige Filter besonders stark reagiert hat, da hier besonders hohe Aktivierungswerte entstanden sind (siehe Abbildung 4-3, S. 35).



Abbildung 4-3: Visualisierung der Filter des Convolutional-Layers und der entstehenden Feature-Maps für ein Inputbild mit der Ziffer 5

Für die Visualisierung wurde die *colormap gray* von *matplotlib* ausgewählt (siehe Abbildung 4-4), wodurch der niedrigste Wert schwarz und der höchste Wert weiß dargestellt wird. Die dazwischenliegenden Werte bekommen einen entsprechenden Graustufenwert zugewiesen.

Diese *colormap* wurde einerseits gewählt, da die Inputbilder für das MNIST-CNN-Modell Graustufenbilder sind und es dadurch eine einheitliche Darstellung und einen direkten Bezug zum Inputbild gibt und andererseits, da die Helligkeitsabstufung zwischen schwarz und weiß ohne weitere Erläuterung auf einen Blick ersichtlich erscheint, im Gegensatz zu *colormaps*, die einen Verlauf von einer Farbe zu einer anderen darstellen.



Abbildung 4-4: Colormap gray von matplotlib

Als Zusatz zu den Ansätzen von Irvine und Chollet wurde der Funktion *imshow* über die Parameter *vmin* und *vmax* jeweils ein fester Minimal- und Maximalwert übergeben. Ohne diese Angabe wird für jedes Bild automatisch der niedrigste und höchste vorhandene Wert bestimmt und dementsprechend die Farbwerte verteilt. Die Bilder sind somit nicht miteinander vergleichbar, da bei jedem Bild beispielsweise die schwarze Farbe einen anderen Wert repräsentiert, nämlich den niedrigsten Wert für das einzelne Bild. Für die CNN-Visualisierung wurde aus den Aktivierungs- bzw. Filterwerten eines Layers das Minimum

und Maximum bestimmt und dieses für jedes Bild an die Funktion übergeben. Dies wurde angepasst, um eine Vergleichbarkeit zwischen den verschiedenen Feature-Maps bzw. Filtern eines Layers zu ermöglichen. Dadurch kann an der Visualisierung der Feature-Maps beispielsweise auch erkannt werden, welche Filter im Vergleich zueinander besonders hohe Aktivierungswerte produziert haben.

Die Werte der Filter eines Layers können über die Keras-Funktion `get_weights()` abgefragt werden und wurden auf dieselbe Art und Weise visualisiert, wie die Feature-Maps.

Zusätzliche Anpassungen am Code wurden vorgenommen, um jedes Filter- bzw. Feature-Map-Bild einzeln und auf den Bildbereich zugeschnitten abzuspeichern (siehe Abbildung 4-5).

```
for i in range(nb_filters_1):
    fig, ax = plt.subplots();
    plt.imshow(weights_1[:, :, 0, i].reshape(3, 3), cmap='gray',
               interpolation='none', vmin=min_1, vmax=max_1);
    ax.set_xticklabels([]);
    ax.set_yticklabels([]);
    ax.axis('off');
    plt.tight_layout();
    if i < 10:
        plt.savefig("Small_Data/Conv1_Filter/conv1_filter_0"+
                    str(i)+".png", transparent=True, bbox_inches='tight');
    else:
        plt.savefig("Small_Data/Conv1_Filter/conv1_filter_"+
                    str(i)+".png", transparent=True, bbox_inches='tight');
plt.close(fig);
```

Abbildung 4-5: Python-Quellcode zum Speichern der Filterbilder

Neben den Bildern wurden auch die Filterwerte des Modells in einer txt-Datei gespeichert, ebenso wie die *Predictions* für die in der Anwendung verwendeten Inputbilder.

4.4 Implementierung der VR-Anwendung

In den folgenden Abschnitten wird die Implementierung der VR-Anwendung mit Unity beschrieben, unterteilt in die Umsetzung der Visualisierung, die Gestaltung der virtuellen Umgebung und die Interaktionstechnik. Bildschirmaufnahmen aller Bestandteile der Visualisierung können Anhang A entnommen werden.

4.4.1 Umsetzung der Visualisierung

Um auf Machine-Learning-Modelle, die beispielsweise mit TensorFlow erstellt und trainiert wurden, in Unity zugreifen zu können, konnte zuvor das TensorFlowSharp Plugin verwendet werden. Inzwischen soll laut Unity dafür die sogenannte Unity Inference Engine (Unity Technologies, 2019d) genutzt werden, die ab Version 0.7.0 in Unitys ML-Agents Toolkit integriert ist (Shih, 2019). Im Rahmen dieser Arbeit wurde keine Schnittstelle zur Verknüpfung mit einem TensorFlow-Modell implementiert, da der Fokus auf der Umsetzung und Evaluierung des Visualisierungskonzeptes lag. Eine Verknüpfung mit einem Modell, welches in Echtzeit Daten in die Anwendung einspeist, könnte bei der Weiterentwicklung des Prototyps erfolgen. Für den Prototyp wurden die für die Visualisierung benötigten Daten, wie in Abschnitt 4.3 beschrieben, erzeugt und dann manuell dem Assets-Ordner des Unity Projekts hinzugefügt. Im Assets-Ordner eines Unity Projekts befinden sich alle Ressourcen, wie 3D-Modelle, Texturen, Skripte und Szenen, die in der Anwendung genutzt werden.

Die virtuelle Umgebung und das Grundgerüst für die Visualisierung wurden mit den im Unity-Editor verfügbaren Tools zur Erstellung von 3D-Spielwelten und Benutzeroberflächen in einer Szene erstellt (siehe Abbildung 4-6).

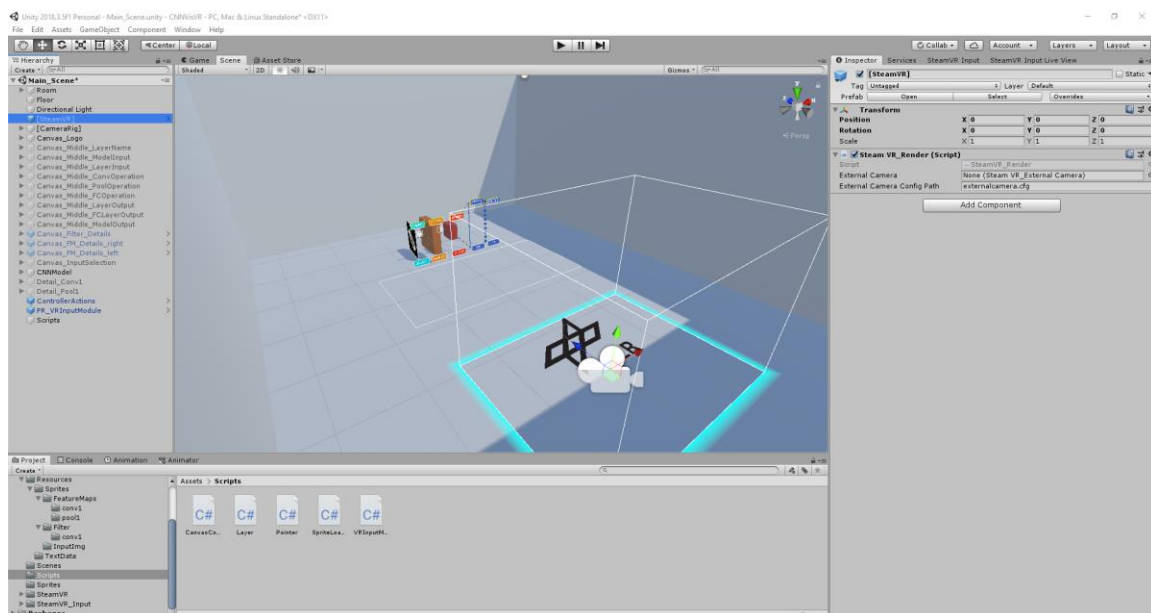


Abbildung 4-6: Ansicht des Unity-Editors mit der Ansicht der Szene

So wurde auch das 3D-Modell mit Beschriftungen und interaktiven Buttons erstellt, das in Abbildung 4-7 zu sehen ist. Zu Beginn der Anwendung ist dieses Modell sichtbar und die weiteren Elemente der Visualisierung erscheinen, sobald der Nutzer einen Layer auswählt.

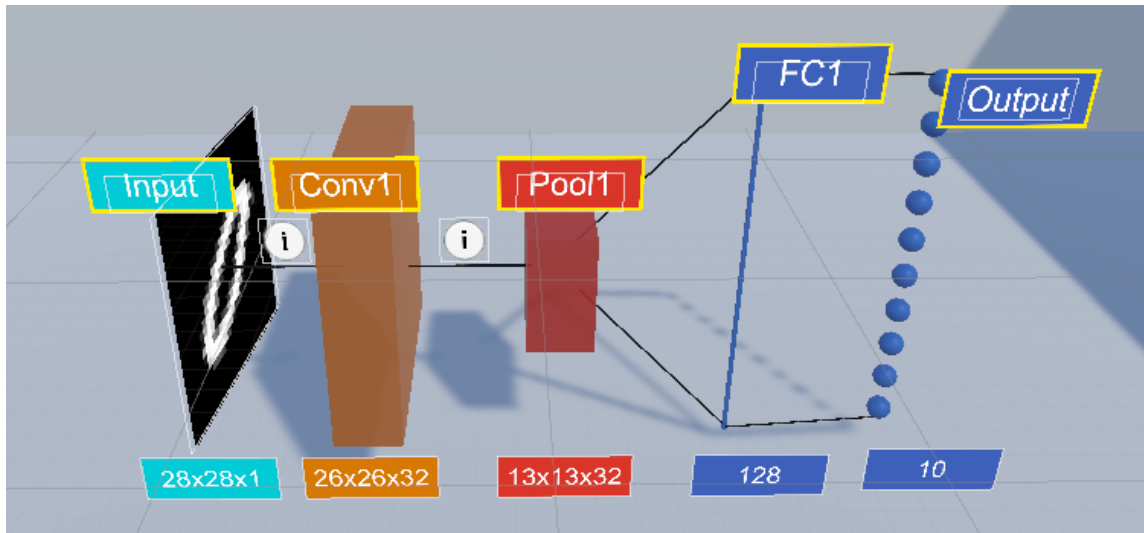


Abbildung 4-7: Das 3D-Modell in der Szene im Unity-Editor

Die Anwendungslogik wurde mit fünf C#-Skripten mit rund 1500 Codezeilen implementiert, welche der Szene hinzugefügt wurden und durch entsprechende Verknüpfungen auf alle Elemente in der Szene zugreifen können. Die Implementierung der Anwendung wird im Folgenden anhand der Skripte erläutert.

SpriteLoader

Die Bilder für Filter und Feature-Maps, sowie die Werte der Filter und die Inputbilder werden über das Skript *SpriteLoader* geladen. Sie sind im Resources-Ordner des Projektes gespeichert. Inhalte des Resources-Ordners können über die Unity-Funktion *Resources.LoadAll<Datentyp> („Ordnerpfad“)* während der Laufzeit geladen werden, damit werden alle Elemente des angegebenen Datentyps geladen, die in dem entsprechenden Ordner gespeichert sind. Im Skript werden sie in Arrays der entsprechenden Datentypen gespeichert, welche dann vom Skript *CanvasController* abgerufen werden können, um sie UI-Elementen zuzuweisen.

Layer

Das Skript *Layer* dient als Blaupause für die Layer des Modells. Hier werden die Attribute *layerID*, *inputLength*, *outputLength*, *layerType* und *layerName* festgehalten. Für jeden Layer des Modells gibt es eine Instanz des Skriptes in der Szene. Das Skript liegt dabei

jeweils auf dem Button des 3D-Modells, der zu dem jeweiligen Layer gehört und die Anzeige der Details zum Layer steuert. Dadurch können bei einer Interaktion mit dem Button die Attribute des entsprechenden Layers abgerufen und dementsprechend die Anzeige der Details auf den Leinwänden angepasst werden.

CanvasController

Das Skript *CanvasController* kontrolliert die Anzeige der Visualisierungselemente. Die Grundlage für die meisten Visualisierungselemente ist dabei jeweils eine sogenannte *Canvas*. Unity bietet eine Vielzahl von vorgefertigten UI-Elementen, um beispielsweise Texte und Bilder anzuzeigen oder die Funktionalität eines Buttons zu bieten. Die UI-Elemente dienen als eine Art Blaupause und können entweder im Editor oder per Skript mit entsprechenden Inhalten gefüllt werden. Alle UI-Elemente, wie Bilder und Texte, müssen einer *Canvas* untergeordnet sein, um angezeigt werden zu können. Die Anzeigen für die Details zu den verschiedenen Layern des Modells ist durch eine Anordnung entsprechender UI-Elemente in der Szene vorbereitet. Das Skript beinhaltet alle Funktionen, die von den Buttons in der Szene aufgerufen werden und steuert somit die Anzeige der entsprechenden Elemente der Visualisierung, sobald der Nutzer damit interagiert. Diese Funktionen sind als *public* definiert, damit die Buttons darauf zugreifen können.

Sobald der Nutzer am 3D-Modell einen Layer auswählt, wird durch die Interaktion mit dem entsprechenden Button die Funktion *ShowLayerDetails()* im Skript *CanvasController* aufgerufen. In dieser Funktion wird zunächst bestimmt, um welchen Typ von Layer es sich handelt und dann abhängig davon die Anzeige der Details des Layers mit weiteren Funktionen vorbereitet und umgesetzt. Bei der Anzeige eines Convolutional-Layers werden beispielsweise die entsprechenden Filter- und Feature-Map-Bilder abgerufen und dann alle UI-Elemente eingeblendet, die zu dem ausgewählten Layer gehören. Auf dem ersten Convolutional-Layer wird als Input das ausgewählte Inputbild angezeigt. Dabei wurde eine Animation eingebaut, die das Bild eines Filters über das Inputbild laufen lässt, um den Prozess darzustellen. Die Animation wurde mit dem *Animator* des Unity-Editors erstellt. Die Visualisierung des Convolutional-Layers zeigen Abbildung 4-8 und Abbildung 4-9 (S. 40).

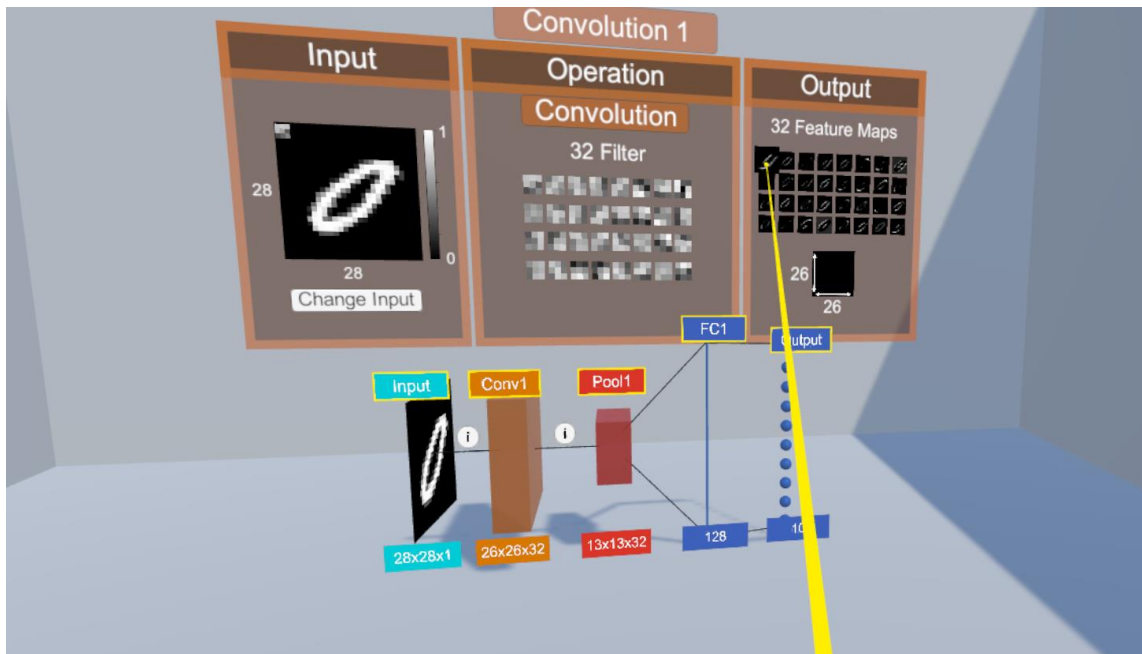


Abbildung 4-8: Auswahl einer Feature-Map auf dem Convolutional-Layer

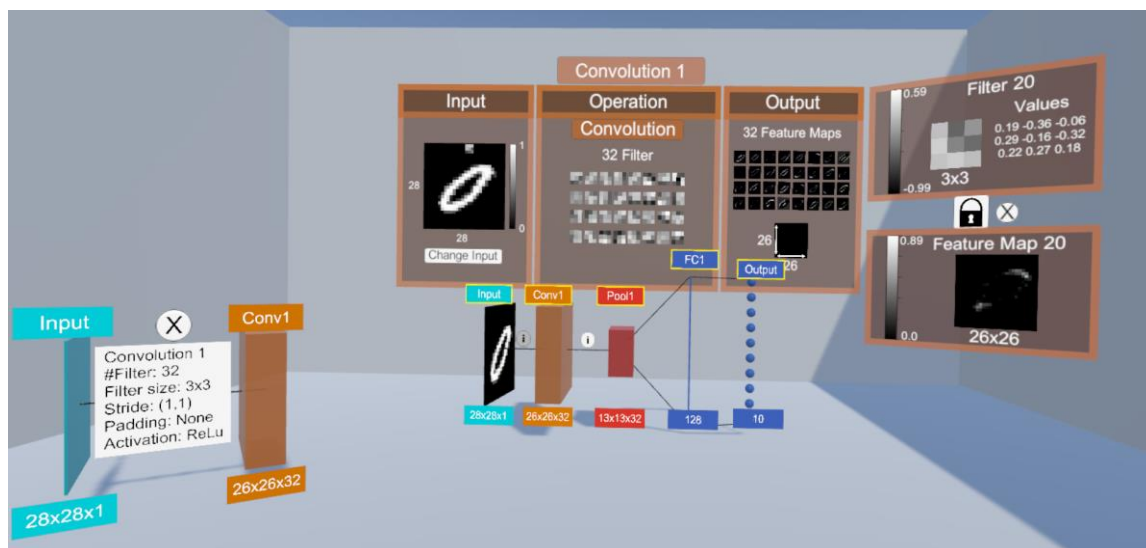


Abbildung 4-9: Überblick über die Visualisierung zum Convolutional-Layer

Die Funktionen *ShowFilterDetails()* und *ShowFMDetails()* werden aufgerufen, wenn mit einem Filter- oder Feature-Map-Bild interagiert wird und steuern die Anzeige ihrer Detailfenster. Die Abbildung 4-10 (S. 41) zeigt eine Bildschirmaufnahme der Anwendung, die eine Übersicht über die Visualisierung zum Pooling-Layer und die verschiedenen Visualisierungselemente bietet.

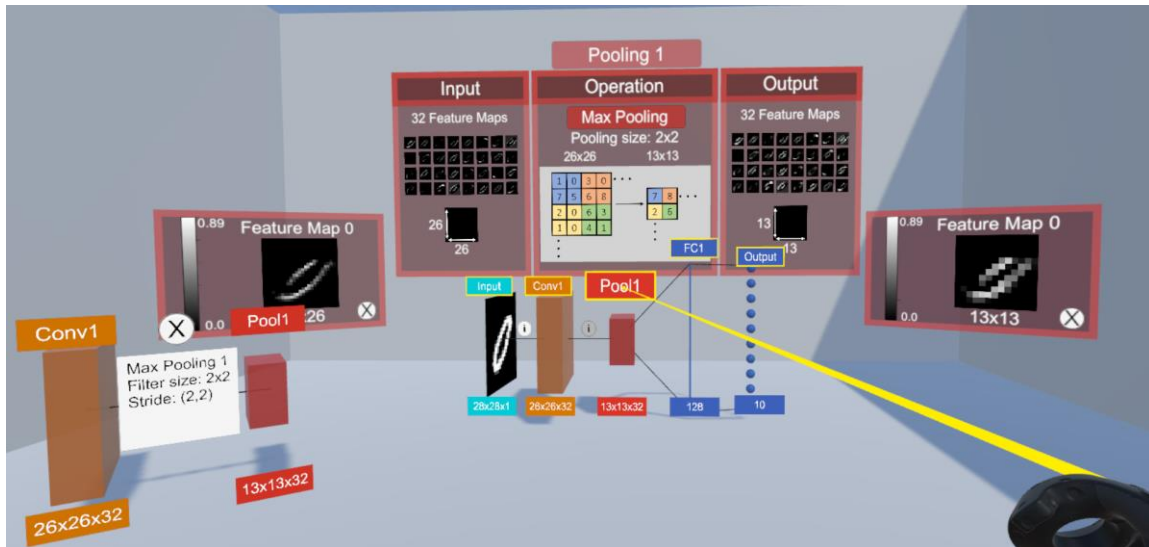


Abbildung 4-10: Übersicht über die Visualisierung zum Pooling-Layer

Des Weiteren beinhaltet das Skript die Funktionen *Hide()* und *HideDetailObj()*, die von den X-Buttons in der Szene aufgerufen werden, um verschiedene Elemente der Visualisierung auszublenden. Die Funktionen *ShowInputSelection()* und *ChangeInputImg()* ermöglichen die Auswahl eines anderen Inputbildes und passen die Inhalte der Visualisierung nach einer Änderung des Inputbildes an. Die Funktion *ShowInfoModel()* ist mit den i-Buttons am 3D-Modell verknüpft, um die kleinen 3D-Modelle mit den Parameterdetails einzublenden. Auch diese wurden bereits in der Szene fertig gestaltet und werden vom Skript ein- oder ausgeblendet. Die Filter- und Feature-Map-Details werden verknüpft angezeigt, sodass ein Filter und die Feature-Map, die daraus entsteht, verglichen werden können. Durch den Schloss-Button kann diese Verknüpfung auch aufgehoben werden. Dieser Button ruft die Funktion *Link()* auf.

Weitere Funktionen, die als *private* definiert sind, werden von den beschriebenen Funktionen aufgerufen, um spezialisierte Aufgaben zu erfüllen und sind im Folgenden aufgelistet.

- `private void ShowInput()`
- `private void ShowConv()`
- `private void ShowPool()`
- `private void ShowFC()`
- `private void ShowOutput()`
- `private void ShowLinkedFilter(int index)`
- `private void ShowLinkedFM(int index)`
- `private void ShowFilterValues(int index)`
- `private void UpdateFilters()`
- `private void UpdateFms()`

Die Skripte **Pointer** und **VRInputModule** ermöglichen die Interaktionen mit dem User Interface und werden in Abschnitt 4.4.3 erläutert.

4.4.2 Gestaltung der virtuellen Umgebung

Bei der Umsetzung des Konzepts als Prototyp wurde der Fokus auf die Implementierung der Visualisierung gelegt, doch auch bei der Gestaltung der virtuellen Umgebung wurden einige wichtig erscheinende Aspekte beachtet, die im Folgenden aufgeführt werden.

Die virtuelle Umgebung wurde als Raum mit vier Wänden und Boden gestaltet und nicht als offene Umgebung, die sich unbegrenzt in alle Richtungen erstreckt, um die räumliche Orientierung für die Nutzer zu verbessern und dadurch dem Auftreten von Symptomen der Reise- bzw. Bewegungskrankheit entgegenzuwirken und die Immersion zu verbessern.

Für die Umgebung wurde die *Default-Skybox* von Unity mit Himmel und Sonne gewählt, da sie ein natürliches Umgebungslicht erzeugt. Auf eine Decke für den virtuellen Raum wurde dabei verzichtet, um das natürliche Licht durch die *Skybox* zu erhalten und dem Nutzer trotz der Wände das Gefühl eines offenen Raumes zu geben, der nicht einengt. Insgesamt wurde die Umgebung sehr hell gestaltet, da bei einem eigenen Test mit hellen und dunklen Umgebungen für die Anwendung die helle Umgebung als freundlicher und

einladender empfunden wurde. Bei Erfahrungen mit einem vorherigen VR-Projekt zeigte sich, dass eine helle Umgebung von den Nutzern als positiv aufgefasst wurde.

Die virtuelle Umgebung wurde abgesehen von dem Raum nicht mit weiteren Objekten ausgestattet, um den Fokus der Nutzer auf die Visualisierung zu lenken. Einen Überblick über die virtuelle Umgebung im Unity-Editor bietet Abbildung 4-11.

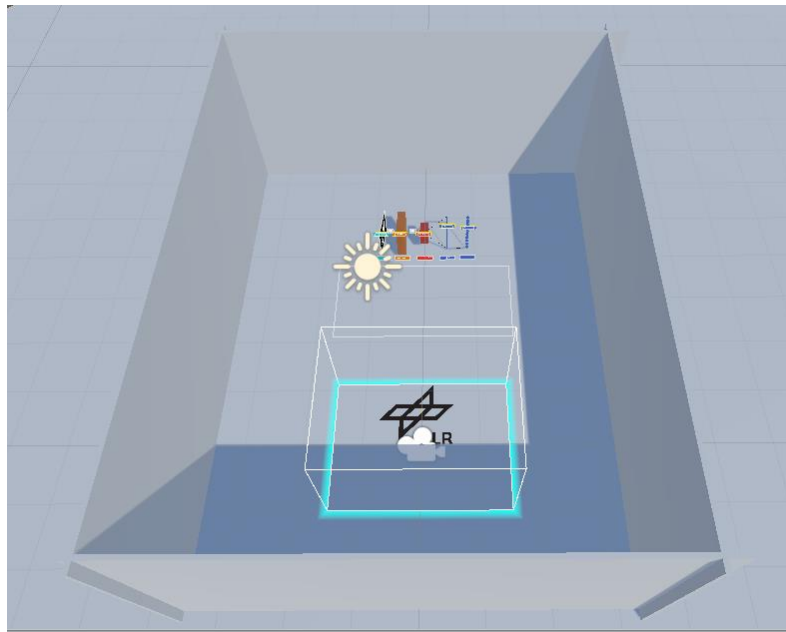


Abbildung 4-11: Überblick über die virtuelle Umgebung im Unity-Editor

Auf dem Boden des virtuellen Raumes wurde das Logo des DLR platziert, um den Nutzern zu Beginn der Anwendung einen Anhaltspunkt für eine gute Platzierung in Bezug auf die Visualisierung zu bieten. Bei der Nutzung der Anwendung können sie sich jedoch im Rahmen des in SteamVR festgelegten Bewegungsbereiches frei bewegen. Die virtuelle Umgebung ist dabei so aufgebaut, dass der Nutzer nicht direkt bei der Visualisierung stehen kann, sondern in Unity Einheiten ein paar Meter vom 3D-Modell entfernt steht, damit er einen guten Überblick über die Visualisierung hat. Diese Maße können jedoch nicht direkt auf die Maße in der realen Umgebung übertragen werden.

4.4.3 Interaktionstechnik

Die Interaktionstechnik, die im Konzept in Abschnitt 3.4 beschrieben wird, wurde anhand eines Tutorials implementiert. Der Quellcode für das Skript **Pointer** stammt von (VR with Andrew (Andrew Connell), 2019a) und für das Skript **VRInputModule** von (VR with Andrew (Andrew Connell), 2019b), welche Inhalt einer Tutorialreihe sind.

Es wurde ein Pointer erstellt, der mit dem Controller verbunden ist. Der Controller ist als 3D-Modell für den Nutzer in der virtuellen Umgebung sichtbar (siehe Abbildung 4-12).

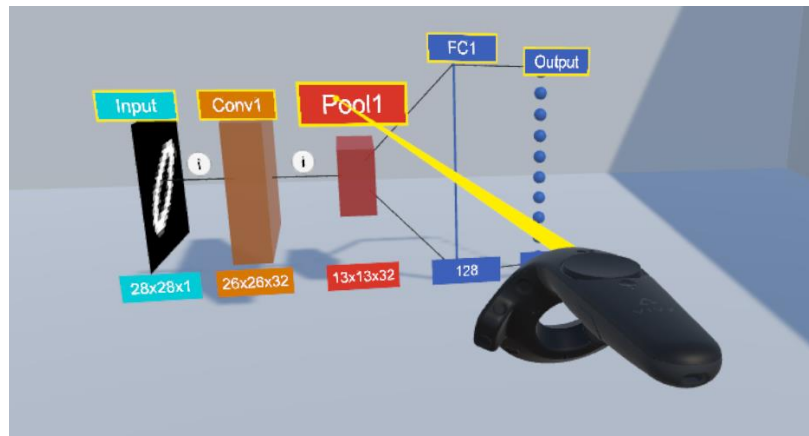


Abbildung 4-12: Interaktion mit dem 3D-Modell

In der Szene wurde ein Objekt namens PR_Pointer erstellt, das in der Hierarchie dem rechten Controller untergeordnet ist. Der PR_Pointer hat eine *Line-Renderer*-Komponente, mit welcher der Pointer-Strahl visuell erzeugt wird, eine *Camera*-Komponente, damit das Objekt als *Eventcamera* genutzt werden kann und das Skript **Pointer**. In dem Pointer-Skript wird auf die *Line-Renderer*-Komponente zugegriffen, um die Länge des Pointer-Strahls festzulegen. Die Länge des Pointer-Strahls wird normalerweise auf die Distanz zum nächsten Objekt in der Szene mit *Collider*-Komponente festgelegt. Dadurch entsteht für den Nutzer der Eindruck, dass der Pointer auf eine feste Oberfläche trifft, die er nicht durchdringen kann. Damit der Pointer-Strahl nicht unendlich lang wird, sobald es keinen *Collider* in seinem Weg gibt, wird eine Standardlänge festgelegt. Die Kollision mit Objekten in der Szene wird über Raycast ermittelt. Am Ende des Pointer-Strahls ist ein 3D-Objekt des Typen *Sphere* befestigt, dessen Position sich der Länge des Pointer-Strahls anpasst. Dadurch ist das Ende des Pointers für den Nutzer deutlicher zu erkennen.

Alle UI-Elemente, mit denen interagiert werden kann, sind vom Typ Button, befinden sich auf einer *Canvas* und bringen bereits die nötigen Eigenschaften mit, damit mittels Mausklick mit ihnen interagiert werden kann. Diese Interaktion ist durch die *Eventcamera*-Komponente möglich, die jeder *Canvas* automatisch zugewiesen ist. Zusammen mit dem *Eventsystem* in der Szene kann dadurch ein Mausklick auf der Position eines Buttons registriert und mittels eines Events verarbeitet werden. Damit die Buttons auf eine Interaktion mit dem Pointer mittels Raycast reagieren, müssen die *Eventcamera* und das

Eventsystem entsprechend angepasst werden. Als neue *Eventcamera* wird das PR_Pointer-Objekt verwendet. Es muss jeder *Canvas*, die Interaktionen mit dem Pointer registrieren soll, als *Eventcamera* zugewiesen werden. Durch das Objekt PR_VRInputModule in der Szene, mit dem Skript **VRInputModule** und dem *EventSystem* Skript von Unity als Komponenten, wird das *EventSystem* angepasst, damit es auf den Controllerinput in Verbindung mit dem Pointer reagiert. In dem Skript wird auf Komponenten von SteamVR zugegriffen. Mit einer Variable vom Typ *SteamVR_Input_Sources* kann der rechte Controller als Target Source definiert werden, also als Quelle aus der die zu verarbeitende Interaktion kommt. Durch eine Variable vom Typ *SteamVR_Action_Boolean* kann definiert werden, auf welche Art der Interaktion mit dem Controller reagiert werden soll. Für die Verarbeitung der Inputs von Controllern von verschiedenen VR-Systemen bietet SteamVR sogenannte Action Sets an, in denen verschiedene Actions definiert werden können. Für diese Anwendung wurde die Action *Teleport* des Action Sets *default* genutzt, worüber Interaktionen mit dem Touchpad des Controllers abgefragt werden können. Das Action Set muss beim Start der Anwendung geladen werden. Dafür ist das Skript von SteamVR *SteamVR_ActivateActionSetOnLoad* zuständig, welches sich auf dem ControllerActions-Objekt in der Szene befindet.

Die Farbänderung der Buttons, die normalerweise den Status des Buttons, *normal*, *highlighted*, *pressed* oder *disabled*, angibt, wurde durch Animationen ersetzt. Die Animationen wurden mithilfe des im Unity-Editor integrierten *Animators* erstellt. Wenn der Nutzer mit dem Pointer über einen Button fährt, wird dessen Größe skaliert, um zu signalisieren, dass mit ihm interagiert werden kann. Beim Drücken des Touchpads am Controller, an dem der Pointer befestigt ist, wird die Interaktion des Buttons ausgeführt.

5 Evaluierung

Im Rahmen der Masterarbeit wurde eine Evaluierung des Prototyps der Anwendung durchgeführt. Das Ziel der Evaluierung war es, einen ersten Eindruck davon zu bekommen, wie die Anwendung von potenziellen Nutzern aufgenommen wird. Der Fokus lag darauf, zu erfassen, inwieweit die Visualisierung zum Verständnis von CNNs beitragen kann und wie der Einsatz von Virtual Reality für die Visualisierung von den Nutzern beurteilt wird. Zusätzlich sollte die Usability der Anwendung evaluiert werden und welche Verbesserungs- und Weiterentwicklungsmöglichkeiten sich für den Prototyp anbieten.

In diesem Kapitel werden die Methodik und Planung sowie der Ablauf der Evaluierung und die anschließenden Ergebnisse erläutert. Am Ende des Kapitels folgt die Auswertung der Evaluierung mit Handlungsempfehlungen für weitere Evaluierungen und die Weiterentwicklung des Prototyps.

5.1 Methodik

Es wurde eine überwiegend qualitative Evaluierung gewählt, da dadurch nicht nur reine Messdaten erhoben werden können, sondern beispielsweise bei Befragungen auch Begründungen für abgegebene Bewertungen erfasst werden können (Bortz & Döring, 2006, S. 297). Döring und Bortz (2006, S. 297) schreiben, dass qualitatives Material durch mehr Details reichhaltiger erscheint als ein Messwert. Dies erschien für die Evaluierung des Prototyps sinnvoll, da dadurch die Möglichkeit besteht die Bewertungen und Einschätzungen der Teilnehmer besser zu verstehen und aus den Begründungen Verbesserungsmöglichkeiten für den Prototyp abzuleiten. Für die Datenerhebung wurde daher ein Fragebogen verwendet, der neben quantitativen Fragen zur Zusammensetzung des Teilnehmerkreises und Einschätzungen zu verschiedenen Bestandteilen der Anwendung auch offene Fragen beinhaltete. Das offene Vorgehen ermöglicht das „Aufdecken vorher nicht erwarteter Aspekte“ (Döring & Bortz, 2016, S. 26). Durch die offenen Fragen sollten die Teilnehmer daher in ihren Antworten möglichst wenig eingeschränkt werden, um viele Gedanken zur Visualisierung festhalten zu können, damit daraus neue Ansätze für die Weiterentwicklung des Prototyps gewonnen werden können. Da der Teilnehmerkreis ausschließlich aus Mitarbeitern des DLR bestand, die in einer Abteilung mit starkem Informatikbezug arbeiten, in der öfter Evaluierungen vorgenommen werden, wurde davon ausgegangen, dass die

offenen Fragen zu verwertbaren, weiterführenden Ergebnissen führen könnten. Als zweite Methode zur Datenerhebung wurde eine Beobachtung gewählt, um Auffälligkeiten bei der Bedienung der Anwendung festzuhalten. Der Fragebogen und die Beobachtung werden im Folgenden beschrieben.

5.1.1 Fragebogen

Der Fragebogen wurde zusammengestellt, um die folgenden Daten zu erheben:

- Angaben zur Zusammensetzung des Teilnehmerkreises
 - Alter & Geschlecht
 - Mögliche Beeinträchtigungen bei der Nutzung der VR-Anwendung, wie das Tragen einer Brille, eine Rot-Grün-Sehschwäche oder Beeinträchtigung der Tiefenwahrnehmung
 - Vorwissen der Personen in Bezug auf Maschinelles Lernen, Künstliche Neuronale Netze und Convolutional Neural Networks
- Einschätzungen zum Wissensstand zu CNNs während verschiedener Abschnitte der Evaluierung
- Einschätzungen zu
 - der Usability der Anwendung
 - besonders unterstützenden und nicht unterstützenden Elementen der Visualisierung
 - der Verständlichkeit der Visualisierung
 - dem Einsatz von Virtual Reality bei der Visualisierung
- Sonstige Anmerkungen zur Anwendung und/oder der Evaluierung

Der vollständige Fragebogen zusammen mit den erhobenen Antworten kann Anhang B entnommen werden.

Der Fragebogen besteht im Bereich der Angaben zur Zusammensetzung des Teilnehmerkreises, mit Ausnahme der Angabe des Alters, aus geschlossenen Fragen mit vorgegeben Antwortmöglichkeiten. Die Einschätzung der Teilnehmer zu ihrem Vorwissen zu den Themen Maschinelles Lernen, Künstliche Neuronale Netze und Convolutional Neural Networks wurde mittels siebenstelliger Likert-Skalen abgefragt. Die Verwendung der Likert-Skala wurde gewählt, da sie sich für die Erhebung von Einstellungen eignet (Likert, 1932)

und durch den Fragebogen die Einstellungen der Teilnehmer zu bestimmten Aspekten der Anwendung erhoben werden sollten. Dabei wurde jeweils danach gefragt, wie gut sich die Teilnehmer mit dem jeweiligen Thema auskennen. Die Likert-Skala ist durch Zahlen von eins bis sieben beschriftet, wobei der Skalenwert eins die Beschriftung „1: Gar nicht“ und der Skalenwert sieben die Beschriftung „7: Sehr gut“ trägt, um das Minimum und das Maximum der Skala eindeutig zu definieren. Diese Skala wurde auch genutzt, um die Einschätzung des Wissensstandes zu CNNs an zwei weiteren Stellen während der Evaluierung abzufragen und so die Änderung des Wissensstandes erheben zu können.

Die anderen Teile des Fragebogens setzen sich aus geschlossenen Fragen, ebenfalls unter Verwendung von Likert-Skalen, und offenen Fragen zusammen. Die Likert-Skalen wurden gewählt, um die Einstellungen der Teilnehmer zur Usability der Anwendung und der Verwendung von Virtual Reality für die Visualisierung abzufragen. Auch bei diesen Fragen sind die Skalen siebenstellig und mit den Zahlen eins bis sieben beschriftet, wobei der erste und letzte Skalenwert zur genaueren Definition jeweils eine zusätzliche Beschriftung besitzt. Der Skalenwert 1 gibt dabei immer das negative Ende und der Wert 7 das positive Ende der Skala an. Darüber hinaus wurden einige offene Fragen gewählt, zum Großteil zur Spezifizierung einer vorangestellten Frage. So wurde beispielsweise in einer geschlossenen Frage in der Ja-Nein-Form danach gefragt, ob die Teilnehmer bestimmte Elemente der Visualisierung schlecht erkennen konnten und bei der Angabe von „Ja“ wurden sie gebeten das Element / die Elemente zu benennen. In der gleichen Form wurde nach offengebliebenen Fragen zu CNNs gefragt und Bestandteilen der Visualisierung, die das Verständnis der Funktionsweise des CNNs nicht unterstützt haben. Weitere offenen Fragen bezogen sich auf Elemente der Visualisierung, die das Verständnis besonders unterstützt haben und auf eine Begründung für die angegebene Einschätzung, inwieweit Virtual Reality die Visualisierung unterstützt hat. Zum Abschluss des Fragebogens folgte eine offene Frage, um die Möglichkeit zur Angabe von weiteren Anmerkungen zur Evaluierung oder der Visualisierung zu geben.

Der Fragebogen wurde mit der Open-Source-Umfrage-Software *LimeSurvey* implementiert, die auf einem Server des DLR läuft. Das Tool ermöglicht die Auswahl von verschiedenen Frage- und Antworttypen und die Verknüpfung von Fragen durch Bedingungen, nach denen sie angezeigt werden sollen. Die Antworten können in verschiedenen Formaten

exportiert werden und dadurch beispielsweise mit Microsoft Excel weiter ausgewertet werden.

Es wurde sich bewusst dafür entschieden den Fragebogen am PC ausfüllen zu lassen, da die Teilnehmer dadurch bei den offenen Fragen zu ausführlicheren Antworten motiviert werden sollten.

5.1.2 Beobachtung

Im Verlauf der Nutzung der Anwendung durch die Teilnehmer wurde eine Beobachtung durchgeführt. Beobachtet wurde dabei primär das Geschehen in der virtuellen Welt, das auf dem Monitor des PCs dargestellt wurde, der die VR-Anwendung ausführte. Dadurch konnte nachvollzogen werden, welche Bereiche der virtuellen Umgebung der Teilnehmer gerade sieht und welche Interaktionen er ausführt.

Die Teilnehmer wurden im Vorfeld über die Beobachtung informiert, es wurde also eine offene Beobachtung (Döring & Bortz, 2016, S. 329) durchgeführt. Während der Beobachtung wurden handschriftliche Notizen angefertigt, es wurden keine Ton- oder Videoaufnahmen der Teilnehmer gemacht.

Durch die Beobachtung sollten Auffälligkeiten bei der Bedienung der Anwendung festgehalten werden, die eventuell nicht durch den Fragebogen erfasst werden. Das Ziel war es durch die Beobachtung des Umgangs der Teilnehmer mit der Anwendung Aspekte zu erfassen, die zu einer Verbesserung der Anwendung beitragen können, sowohl in Bezug auf die Informationsvermittlung als auch in Bezug auf die Bedienung der Anwendung. Es wurde daher eine qualitative, teil-strukturierte Beobachtung durchgeführt (Döring & Bortz, 2016, S. 328). Dafür wurden während der Beobachtung Kommentare der Teilnehmer und Auffälligkeiten während der Nutzung der Anwendung notiert, die sich auf die Fragen bezogen, ob es Verständnis- oder andersartige Probleme in Bezug auf die Bedienung der Anwendung oder die dargestellten Informationen gibt und ob bestimmte Teile der Visualisierung besonders positiv von den Teilnehmern aufgenommen werden.

Die Teilnehmer wurden zu Beginn darauf hingewiesen, dass sie Auffälligkeiten oder Fragen während der Nutzung der Anwendung kommunizieren können, da dies bei der Verbesserung der Anwendung helfen kann. Auf Verständnisfragen wurde auch mit entsprechenden Erklärungen eingegangen und der Vorgang entsprechend in den Notizen

festgehalten. Somit wurde eine teilnehmende Beobachtung mit einer passiven Teilnahme durchgeführt (Döring & Bortz, 2016, S. 329). Es wurde zwischenzeitlich mit dem Teilnehmer kommuniziert, der Hauptfokus lag aber auf der Beobachtung.

Einige Teilnehmer kommentierten die Anwendung nach der Nutzung der Anwendung, was ebenfalls in den Beobachtungsnotizen unter „Kommentare der TP nach Anwendung“ festgehalten wurde. Vereinzelt wurden während dem Ausfüllen des Fragebogens Kommentare von Teilnehmern gemacht, die ebenfalls den Beobachtungsnotizen hinzugefügt wurden. Zusätzlich wurde die Dauer der gesamten Evaluierung notiert und die Zeit, die mit der Nutzung der Anwendung verbracht wurde. Die Beobachtungsnotizen sind im Anhang C beigefügt.

5.1.3 Einführungstext

Es wurde ein kurzer Einführungstext zum Thema Maschinelles Lernen/Künstliche Neuronale Netze/Convolutional Neural Networks erstellt (siehe Anhang D). Dieser wurde vor der Nutzung der Anwendung von den Teilnehmern gelesen, da die Anwendung nicht dafür vorgesehen ist, ohne andere Bezugsquellen eine Einführung in alle drei Themenbereiche zu liefern, sondern speziell CNNs erläutern soll.

5.2 Vorbereitung & Ablauf der Evaluierung

Im Vorfeld der Evaluierung wurden die Teilnehmer über einen DLR internen E-Mail Verteiler eingeladen. Dabei wurden sie darüber informiert, dass es sich um eine Evaluierung im Rahmen dieser Masterarbeit handelt und eine VR-Visualisierung evaluiert wird, die ein Convolutional Neural Network zur Bilderkennung darstellt und deren Ziel es ist den Einstieg in das Thema der Künstlichen Neuronalen Netze und speziell CNNs zu begleiten. Der Teilnehmerkreis der Evaluierung bestand aus 14 Mitarbeitern der DLR-Einrichtung Simulations- und Softwaretechnik in Köln. Die Teilnahme war freiwillig.

Die Evaluierung fand beim DLR am Standort in Köln-Porz statt. Dort gibt es in der Abteilung Intelligente und Verteilte Systeme einen Raum, der als VR-Labor dient und mit entsprechenden PCs und einem Aufbau mit Basisstationen für die HTC Vive Pro ausgestattet ist.

Der PC für die Evaluierung hatte die folgende Ausstattung:

- Betriebssystem: Windows 7 Enterprise, 64-Bit
- Prozessor: Intel® Xeon® CPU E5645 @ 2.40GHz
- RAM: 24,0 GB
- Grafikkarte: NVIDIA Geforce GTX 1080, 8,0 GB

Es wurden die SteamVR Version 1.2.10 genutzt und die Anwendung wurde mit Unity in Version 2018.3.5f1 ausgeführt. Die Teilnehmer nutzten die VR-Anwendung mit der HTC Vive Pro. Für die Evaluierungs-Termine wurden jeweils 30-45 Minuten für eine Person angesetzt. Der Ablauf wurde wie folgt gestaltet:

- Begrüßung und Erläuterung des Ablaufs
- Ansicht und Unterschrift der Datenschutzerklärung durch Teilnehmer (siehe Anhang E)
- Ausfüllen des 1. Fragebogenteils (Allgemeine Angaben, Vorwissen)
- Lesen des Einführungstexts und Einschätzung des Wissensstandes zu CNNs nach dem Text
- Einführung in die VR-Anwendung
- Nutzung der VR-Anwendung durch Teilnehmer mit Beobachtung
- Ausfüllen des 2. Fragebogenteils
- Ende der Evaluierung

Die Zeit in der Anwendung wurde mittels einer Stoppuhr-Anwendung auf einem Smartphone festgehalten und in den Beobachtungsnotizen notiert. Die Aufgabenstellung der Teilnehmer während der Nutzung der Visualisierung lautete die Visualisierung zu erkunden und zu versuchen die Funktionsweise des CNNs zu verstehen.

5.3 Ergebnisse der Evaluierung

Für die Auswertung der Evaluierung wurden sechs Kategorien festgelegt, denen die erhobenen Daten, sowohl des Fragebogens als auch der Beobachtungen, zugeordnet wurden. Wenn die erhobenen Daten es zugelassen haben, wurden weitere Unterkategorien gebildet, unter denen die Ergebnisse zusammengefasst wurden.

Es folgt eine Auflistung der Kategorien:

Zusammensetzung des Teilnehmerkreises: Wie sich der Teilnehmerkreis der Evaluierung zusammensetzt, z. B. Alter, Geschlecht, Vorwissen

Wissenserwerb: Inwieweit das Verständnis von CNNs durch die Visualisierung verbessert werden konnte und Verbesserungsmöglichkeiten in Bezug auf die Informationsvermittlung

Einsatz von Virtual Reality: Inwieweit der Einsatz von Virtual Reality die Visualisierung unterstützt hat und Verbesserungsmöglichkeiten zum Einsatz von Virtual Reality

Usability der Anwendung: Inwieweit die Anwendung benutzerfreundlich war und mögliche Verbesserungen in Bezug auf die Usability

Erweiterung der Anwendung: Möglichkeiten, um die Anwendung im Funktionsumfang zu erweitern

Auffälligkeiten in Bezug auf das Vorwissen: Auffälligkeiten bei den Ergebnissen in Bezug auf das Vorwissen der Teilnehmer

Die Daten, die mit dem Fragebogen und durch die Beobachtung erhoben wurden, sind im Anhang aufgeführt (siehe Anhang B und Anhang C). Die Tabelle 5-1 zeigt die Zuordnung der Fragen des Fragebogens zu den Kategorien. Die Fragencodes können den Fragen im Anhang zugeordnet werden.

Kategorie	Fragencode(s)
Zusammensetzung des Teilnehmerkreises	F01- F12
Wissenserwerb	F12-F13, F20-F21a, F25-F27
Einsatz von VR	F23-F24a
Usability	F14-F19a, F21a-F22, F27
Erweiterung der Anwendung	F25a, F27

Tabelle 5-1: Zuordnung der Fragen des Fragebogens zu den Oberkategorien

Teilweise wurden offene Fragen mehreren Kategorien zugeordnet, da einzelne Antworten besser zu einer anderen Kategorie passen. So wurden beispielsweise Antworten zu Frage F21a nach Elementen, die das Verständnis nicht unterstützt haben, teilweise der Kategorie

Usability zugeordnet, statt der Kategorie Wissenserwerb, da sich diese Antworten eher auf die Bedienung als auf das übermittelte Wissen bezogen. Nach dieser Einordnung in die Kategorien wurden die Antworten der offenen Fragen, wenn möglich, erneut in Unterkategorien zusammengefasst, um zu jeder Frage Aussagen zu generieren, die von den Antworten mehrerer Personen abgeleitet wurden.

Die Beobachtungsnotizen wurden ebenfalls in die vier Kategorien Wissenserwerb, Einsatz von VR, Usability und Erweiterung der Anwendung eingeteilt. Die Unterkategorien gelten teilweise sowohl für Antworten auf offene Fragen, als auch für Daten der Beobachtungen, wenn diese zusammengefasst werden konnten.

Jedem Teilnehmer wurde eine ID zugeordnet, die sowohl den Fragebogen als auch die entsprechende Beobachtung markiert, damit diese miteinander in Bezug gesetzt werden können. Diese ID besteht aus einem Großbuchstaben und steht weder mit den Angaben des Teilnehmers noch mit der Reihenfolge, in der die Daten erhoben wurden, in Verbindung. Für die Zuordnung von Referenzen zu einer Beobachtung oder einem Fragebogen ist den IDs bei Verweisen auf die Daten ein B für Beobachtung oder ein F für Fragebogen vorangestellt. Falls Antworten des Fragebogens durch den Text nicht eindeutig einer Frage zugeordnet werden, wird der Fragencode der Referenzangabe vorangestellt (Beispiel: F27, W).

5.3.1 Zusammensetzung des Teilnehmerkreises

Der Teilnehmerkreis bestand zu 71,43 % aus weiblichen Personen und zu 28,57 % aus männlichen Personen (F02). Das durchschnittliche Alter lag bei 28,29 Jahren (F01). Das minimale angegebene Alter betrug dabei 22 Jahre, das maximale Alter wurde mit 38 Jahren angegeben. Keiner der Teilnehmer gab an eine Rot-Grün-Sehschwäche zu haben (F03) und nur ein Teilnehmer gab an Probleme mit der Tiefenwahrnehmung zu haben (F04). Dieser Teilnehmer gab jedoch an damit bisher bei VR-Anwendungen keine Probleme festgestellt zu haben (BU). Drei Teilnehmer (21,43 %) gaben an, immer eine Brille zu tragen und sechs (42,86 %) gelegentlich eine Brille zu tragen (F05). Fünf Teilnehmer (35,71 %) gaben an keine Brille zu tragen. Die Teilnehmer, die immer eine Brille tragen und ein Teilnehmer, der die Brille gelegentlich trägt, trugen diese auch während sie die VR-Anwendung genutzt haben (F06). Bei der Evaluierung konnte keine Beeinträchtigung durch das Tragen einer Brille festgestellt werden.

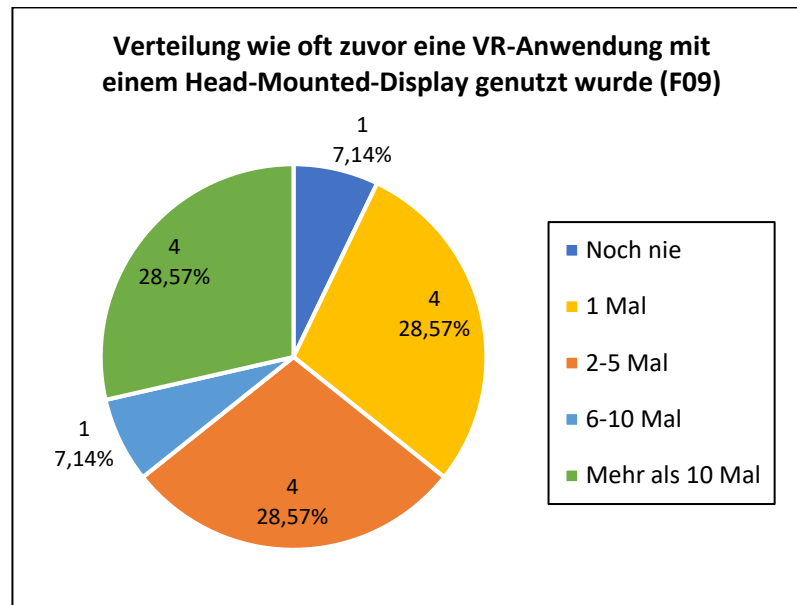


Abbildung 5-1: VR-Erfahrung der Teilnehmer

Bei der VR-Erfahrung (F09, siehe Abbildung 5-1) bildeten sich drei gleich große Gruppen, bestehend aus jeweils vier Personen (28,57 %), die angegeben haben einmal, 2-5 Mal und mehr als 10 Mal eine VR-Anwendung genutzt zu haben. Jeweils eine Person gab an noch nie oder 6-10 Mal eine VR-Anwendung genutzt zu haben.

Vorwissen

Um das Vorwissen der Teilnehmer einzuschätzen wurde zuerst abgefragt, ob sie im Bereich der Informatik und im Bereich des Maschinellen Lernens arbeiten (F07, F08). Eine große Mehrheit der Teilnehmer (11 / 78,57 %) gab an im Bereich der Informatik zu arbeiten. Nur zwei Teilnehmer (14,29 %) gaben an auch im Bereich des Maschinellen Lernens zu arbeiten. Als zweiter Teil der Einschätzung des Vorwissens wurde auf einer 7-stelligen Likert-Skala von „1: Gar nicht“ bis „7: Sehr gut“ abgefragt, wie gut sich die Teilnehmer nach ihrer eigenen Einschätzung mit Maschinellen Lernverfahren, Künstlichen Neuronalen Netzen und Convolutional Neural Networks auskennen (F10-12).

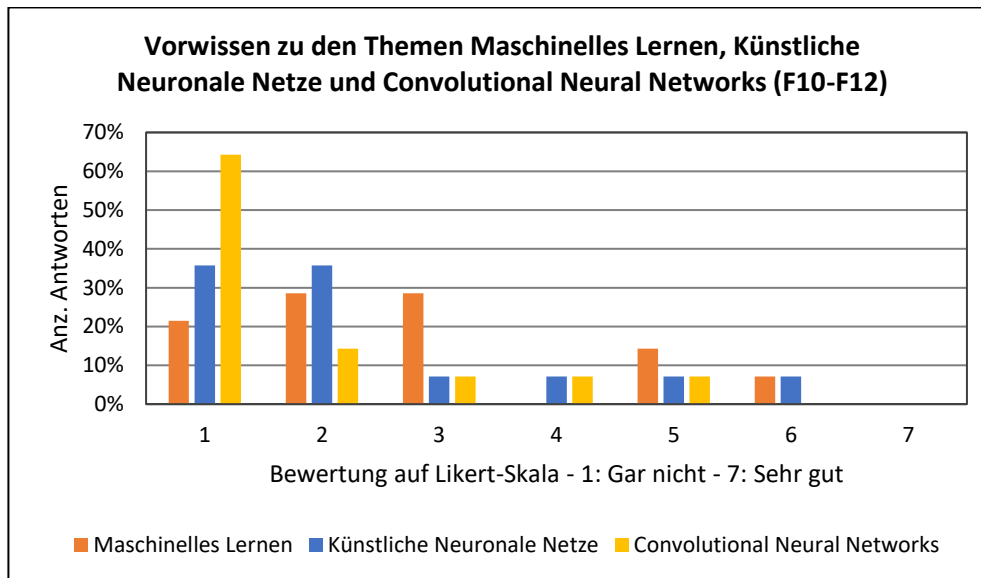


Abbildung 5-2: Vorwissen der Teilnehmer

Bei den Antworten fällt auf, dass die Anzahl der Antwort „1: Gar nicht“ von Frage zu Frage zunimmt (siehe Abbildung 5-2). Gaben bei Maschinellen Lernverfahren nur drei Teilnehmer (21,43 %) an sich gar nicht auszukennen, waren es bei Künstlichen Neuronalen Netzen fünf (35,71 %) und bei Convolutional Neural Networks neun (64,29 %). Bei keiner der drei Fragen gab jemand die höchste Möglichkeit „7: Sehr gut“ an. Bei Maschinellen Lernverfahren und Künstlichen Neuronalen Netzen gaben zwei Teilnehmer (14, 28 %) noch Werte von 4 oder 5 an und eine Person (7,14 %) einen Wert von 6 an. Bei Convolutional Neural Networks gaben nur noch zwei Personen jeweils einen Wert von 4 und 5 an. Bei den beiden ersten Fragen lagen die Angaben einer deutlichen Mehrheit von 11 Personen (78,57 %) in der unteren Hälfte der Likert-Skala, bei der dritten Frage zu Convolutional Neural Networks erhöhte sich diese Zahl auf 12 Personen (85,71 %). Eine Mehrheit der Teilnehmer hatte vor der Evaluierung laut eigener Einschätzung also kein oder sehr wenig Vorwissen zu den drei Themenbereichen, während maximal drei Teilnehmer ihr Vorwissen in einem der Themenbereiche im mittleren bis höheren Bereich einschätzten.

5.3.2 Wissenserwerb

Unter der Kategorie Wissenserwerb sind alle Antworten und Beobachtungen zusammengefasst, aus denen sich Rückschlüsse auf den Wissenserwerb zu CNNs durch die Visualisierung ziehen lassen.

Zwei Personen gaben unter sonstigen Anmerkungen zur Visualisierung/Evaluierung bzw. zu offenen Fragen an, dass sie die Visualisierung für den Einstieg in das Thema von Künstlichen Neuronalen Netzen als hilfreich empfinden (F27, E; F25a, R), wobei eine Person dies damit begründete, dass bei der Visualisierung „der Fokus genau auf die jeweils relevanten Bestandteile gelenkt werden kann“ (F27, E). Eine Person schränkte dabei ein, dass ihrer Meinung nach die Visualisierung allein für die Erlangung eines tiefergehenden Verständnisses nicht ausreicht (F25a, R). Als eine positive Eigenschaft der Visualisierung wurde von einem Teilnehmer hervorgehoben, dass die Visualisierung „fast vollständig selbsterklärend“ (F27, Q) war, was als sehr gut empfunden wurde. Eine Person gab außerdem an, sie denke „das Konzept von CNN verstanden zu haben“ (F27, O).

Weitere Antworten und Beobachtungen gliedern sich in drei Unterkategorien. Als erstes wird die Veränderung des Wissensstandes zu CNNs beschrieben, darauf folgen offene Fragen zu CNNs bzw. Verständnisprobleme bei der Visualisierung und als dritte Unterkategorie werden besonders unterstützende Elemente der Visualisierung aufgeführt.

Veränderung des Wissensstandes

Während der Evaluierung wurde zu drei Zeitpunkten die Einschätzung des Wissensstandes zu CNNs der Teilnehmer abgefragt. Zuerst zu Beginn der Evaluierung im Rahmen der Abfrage des Vorwissens, dann nachdem der Einführungstext gelesen wurde und am Ende, nachdem die Teilnehmer die Anwendung getestet hatten (F12, F13, F26). Damit sollte festgestellt werden, ob sich der Wissensstand zu CNNs durch die Visualisierung tendenziell erhöht. Für jede der Abfragen wurde dabei leicht abgewandelt gefragt, wie gut sich die Teilnehmer mit CNNs auskennen. Für die Antworten wurde für alle drei Fragen die gleiche siebenstellige Likert-Skala verwendet, von „1: Gar nicht“ bis „7: Sehr gut“. Die Abbildung 5-3 zeigt einen Vergleich der Antworten zu den drei Fragen. Daraus wird ersichtlich, dass sich die maximale Anzahl von Antworten von Frage zu Frage auf der Skala weiter nach rechts verschiebt. Zu Beginn lagen die meisten Antworten bei „1: Gar nicht“, nach dem Text bei „2“ und nach der Visualisierung bei „4“. Die Wissensseinschätzung hat sich insgesamt von Frage zu Frage erhöht. Es wurde die Wissensänderung von jedem Teilnehmer zwischen der Frage zu Beginn und nach dem Text und der Frage nach dem Text und nach der Visualisierung berechnet. Die Berechnung ergab, dass zehn Personen eine Verbesserung ihres Wissens von einem Skalenpunkt durch die Visualisierung angeben haben. Zwei

Personen gaben keine Wissensänderung an und eine Person eine Wissensverbesserung von zwei Skalenpunkten. Bei einer Person ergab sich eine Wissensänderung von -1 zwischen der Abfrage nach dem Text und nach der Visualisierung. Es sollte aber wahrscheinlich ein gleichbleibender Wissensstand angegeben werden, da ein entsprechender Kommentar beim Ausfüllen des Fragebogens gemacht wurde (BM). Daraus ergibt sich, dass 78,57 % der Teilnehmer einen Wissenszuwachs zum Thema CNNs durch die Visualisierung angegeben haben.

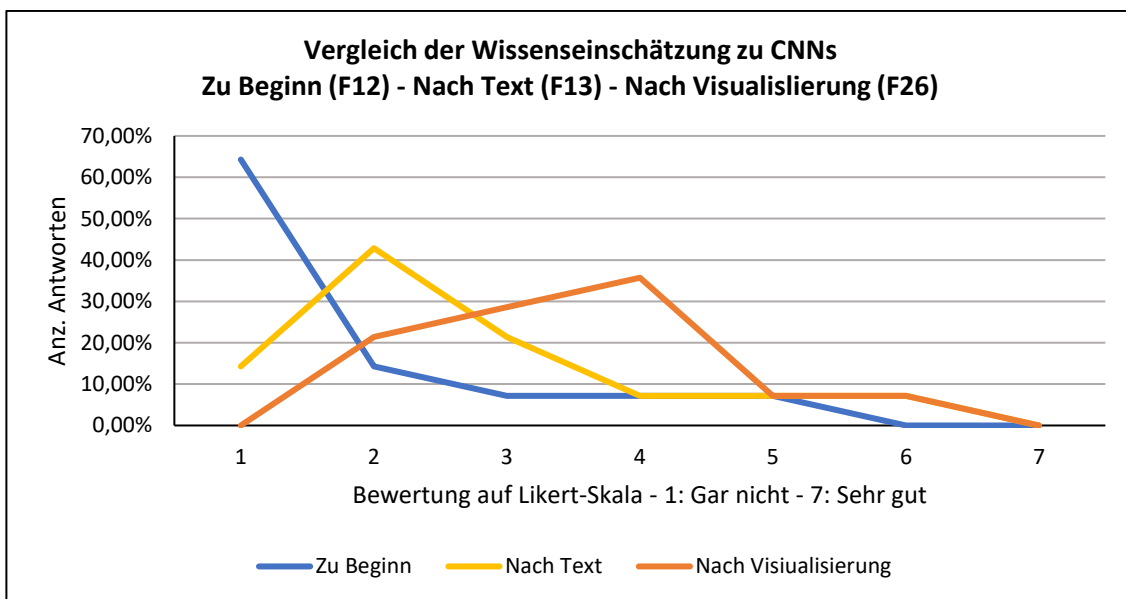


Abbildung 5-3: Vergleich der Wissensseinschätzung zu CNNs

Offene Fragen zu CNNs / Verständnisprobleme

In dieser Unterkategorie sind Evaluierungsergebnisse zusammengefasst, die sich darauf beziehen, inwieweit bei der Visualisierung Verständnisprobleme bestanden und welche Fragen zu CNNs offengeblieben sind und somit aufzeigen in welchen Bereichen die Visualisierung inhaltlich weiterentwickelt werden könnte.

Dazu sind unter dieser Kategorie die Antworten auf die Fragen F21/21a zu Elementen der Visualisierung, die das Verständnis nicht unterstützt oder negativ beeinflusst haben und F25/25a zu bestimmten Fragen zu CNNs, die durch die Visualisierung offengeblieben sind, zusammengefasst. Zu beiden Fragen gibt es Antworten, die besser der Kategorie Usability oder Erweiterung der Anwendung zugeordnet werden können und daher in die entsprechenden Abschnitte einfließen, anstatt hier genannt zu werden (siehe Abschnitt 5.3.4, S. 61 und Abschnitt 5.3.5, S.66).

Acht Personen gaben an, Fragen zu CNNs zu haben, die durch die Visualisierung nicht beantwortet wurden, und sechs Personen gaben an keine Fragen zu haben. Sieben Personen nannten Elemente, die ihr Verständnis nicht unterstützt oder negativ beeinflusst haben und sieben Personen gaben an, dass es solche Elemente nicht gab.

Die zwei wichtigsten Fragen, die aus den Antworten herausgebildet werden konnten, sind:

1. Was bedeuten die Parameter im Detailfenster des kleinen 3D-Modells? (Stride, Padding etc.) (F21a/25a, L / BL; F21a, G; F21a/25a, Z; BW)
2. Wie funktioniert der Lernprozess des CNNs, wie werden Filter gelernt? (F25a, E; F25a, Q; F25a, G)

Des Weiteren wünschten sich zwei Personen mehr Details zu den Berechnungen, die von dem CNN durchgeführt werden, einmal allgemein, ohne einen speziellen Vorgang im CNN zu nennen (F25a, H), und einmal in Bezug auf die Signalverarbeitung des Fully-Connected-Layers (F25a, G).

Zwei Personen gaben an, gerne mehr Details zu den einstellbaren Parametern des CNNs zu haben. Eine Frage richtete sich speziell danach, welche Parameter es für den Fully-Connected-Layer gibt (F25a, G) und die andere richtete sich danach wie genau die Parameter angepasst werden können und welche Auswirkungen dies hat (F25a, Z).

Eine Person gab an, dass es ihrer Ansicht nach viele Daten zu den einzelnen Elementen gab, die „auf die Schnelle nicht in Zusammenhang gebracht werden konnten“ (F21a, H). Eine Person gab mathematische Berechnungen als nicht unterstützendes Element an (F21a, O).

Bei drei Beobachtungen fiel auf, dass die Personen sich die Funktionsweise des CNNs nicht ohne weitere ausführliche Erklärungen erschließen konnten und dementsprechend nachgefragt haben (BO, BA, BW). Bei allen dreien wurde der Vorgang auf dem Convolutional-Layer erklärt, bei zwei von ihnen wurden alle Layer erläutert. Ein wichtiger Hinweis schien dabei zu sein, dass hinter dem Inputbild, den Filtern und den Feature-Maps Zahlenwerte für die einzelnen Pixel stecken, die visualisiert werden (BO, BA, BW). Eine dieser Personen merkte auch im Fragebogen an, dass die Erklärung durch die Beobachterin ihr sehr geholfen hätte und nur durch die Visualisierung nicht alles „sofort klar“ (F27, A) war.

Besonders unterstützende Elemente

Mit der Frage F20 wurden Elemente der Visualisierung abgefragt, die das Verständnis der Funktionsweise des CNNs besonders unterstützt haben. Inhaltlich ähnliche Antworten wurden zusammengefasst und im Folgenden, nach der Anzahl der Nennungen sortiert, aufgelistet:

- **Visualisierung/Veranschaulichung von Daten** (FZ, FW, FO, FR, FQ)
 - Dabei wurden die Filter von zwei Personen explizit genannt (FZ, FO) und von jeweils einer Person der Input (FO) und die Feature-Maps (FQ).
- **Gegenüberstellung von Input und Output allgemein** (FE, FU, FH) **und speziell die Zuordnung von Filter zu Feature-Map** (FS, FE)
- **Übersicht über den Ablauf/die Pipeline durch das 3D-Modell** (FM, FL, FG)
- **Animation des Filters, der sich über das Inputbild bewegt** (FQ, FH)
- **Leinwände mit Details zu den einzelnen Ebenen** (FG, FZ)
 - Dabei wurden von einer Person speziell die Leinwände zu den Filtern und zu Pooling genannt (FG).
- **Visualisierung aller Feature-Maps** (FQ, FY)
- **Komfortables Interface/intuitive Auswahl von einzelnen Teilbereichen** (FY, FM)

Weitere Nennungen von jeweils einer Person:

- Möglichkeit das Verfahren schrittweise mit verschiedenen Inputs durchlaufen zu können (FU)
- Beschriftungen/Überschriften (FO)
- Parameter im Detailfenster des kleinen 3D-Modells (FS)

Auffällig ist, dass von sechs Personen explizit die Darstellung der Filter als besonders unterstützendes Element der Visualisierung genannt wird (FS, FG, FO, FQ, FH, FZ).

5.3.3 Einsatz von Virtual Reality

In dieser Kategorie werden die Ergebnisse in Bezug auf den Einsatz von Virtual Reality für die Visualisierung aufgeführt. Dafür werden die Antworten von drei Fragen ausgewertet und entsprechende Notizen aus den Beobachtungen.

Mit Frage F23 wurde abgefragt, inwieweit die Teilnehmer das Gefühl hatten sich durch den Einsatz von Virtual Reality besser auf die dargestellten Daten konzentrieren zu

können. Mit Frage F24 wurde ihre generelle Einschätzung zum Einsatz von Virtual Reality für die Visualisierung abgefragt. Die durchschnittliche Bewertung zu diesen beiden Fragen kann Abbildung 5-4 entnommen werden. Ein niedriger Skalenwert gibt dabei eine niedrige Bewertung und ein hoher Skalenwert eine hohe Bewertung an. Die Frage F24a schließt an Frage F24 an und fragt nach einer Begründung für die Einschätzung zum Einsatz von VR.

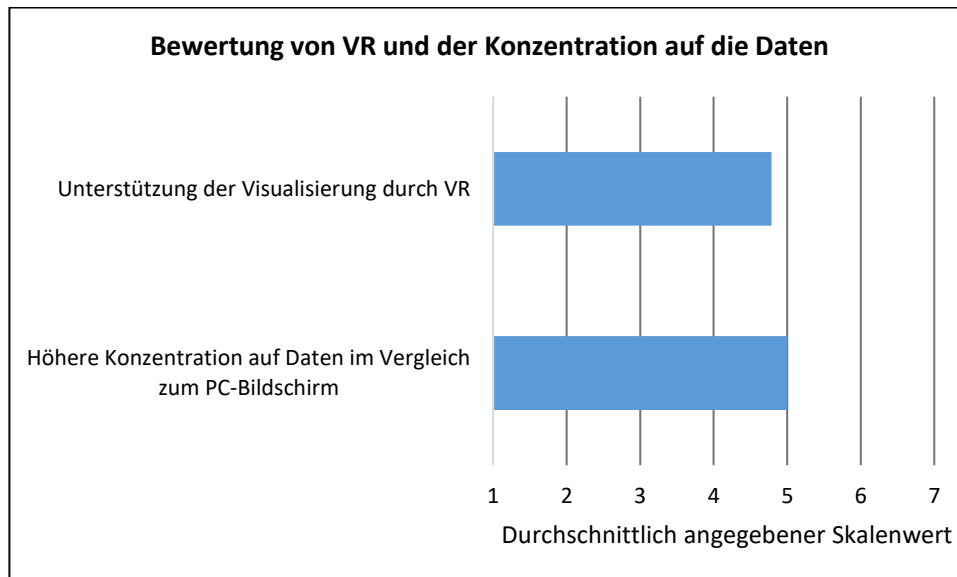


Abbildung 5-4: Bewertung des Einsatzes von VR und der Konzentration im Vergleich zum PC-Bildschirm

Bei beiden Fragen zum Einsatz von Virtual Reality liegen acht, und somit über die Hälfte (57,15 %), der Antworten in der oberen Hälfte der Skala, vier Antworten (28,57 %) in der Mitte und zwei (14,29 %) in der unteren Hälfte der Skala. Bei der Frage zur höheren Konzentration durch VR gibt es dabei einen stärkeren Ausschlag zum positiven Ende der Skala, da 42,86 % der Teilnehmer einen Skalenwert von 6 oder 7 angaben. Bei der Frage, inwieweit VR die Visualisierung unterstützt hat, gaben im Vergleich dazu 28,57 % einen Skalenwert von 6 oder 7 an.

Bei den Begründungen für die Einschätzung zum Einsatz von Virtual Reality im Fragebogen und durch Kommentare während der Beobachtung stellten sich drei Hauptaussagen heraus, die mehrfach getroffen wurden.

Die Hauptaussagen, geordnet nach der Anzahl der Nennungen sind:

1. Die Visualisierung könnte eventuell auch als 2D- oder 3D-Desktop-Anwendung umgesetzt werden (FE/BE, FU, FM/BM, FG, FZ, BY, BQ, 7 Personen)
2. Der Einsatz von Virtual Reality schaltet äußere Ablenkungen aus und führt zu einer höheren Fokussierung auf die Visualisierung (FE, FW, FO, FU, FL, FG, 6 Personen)
3. Die Raumnutzung der Visualisierung ist hilfreich bei der Darstellung der Visualisierung (FQ, FS, 2 Personen)

Als Begründung zur ersten Aussage wurde von drei Personen angegeben, dass bei der Visualisierung die Raumnutzung nicht stark zum Tragen kommt und sie sich die Visualisierung daher auch als PC-Anwendung vorstellen könnten (FU, FE, FZ).

Die weiteren Aussagen zum Einsatz von Virtual Reality lassen sich in drei negative und drei positive Aussagen einteilen.

Als negative Aspekte wurden der umständliche Gebrauch der Ein- und Ausgabegeräte (FQ) und die ungewohnt starke Kopfbewegung zum Umschauen im Gegensatz zu Anwendungen am PC-Monitor genannt (FG). Eine Person gab an, dass der Umgang mit einer VR-Anwendung für sie noch ungewohnt war und sie deshalb teilweise von dem speziellen Umgang mit der VR-Anwendung abgelenkt war (FL).

Als positiver Aspekt wurde die optisch bessere Erkennbarkeit der Daten in VR genannt (FR). Als weiterer positiver Aspekt wurde die Interaktion mit der VR-Anwendung hervorgehoben, die einen einfachen Wechsel zwischen verschiedenen Informationsebenen ermöglicht, wodurch Gedankengänge und Fragestellungen verdeutlicht und besser verständlich wurden (FH). Eine Person gab an, dass der innovative VR-Ansatz die Exploration der Visualisierung spielerischer gestaltet und man dadurch eine höhere Motivation hat, sich damit auseinanderzusetzen (FO).

5.3.4 Usability der Anwendung

Ein Augenmerk der Evaluierung lag auf der Bewertung der Usability der Anwendung durch die Testnutzer. Einerseits wurde die Usability der Anwendung in einigen Fragen des Fragebogens erfasst (F14-25a, F22), andererseits wurden Auffälligkeiten in der Bedienung bei der Beobachtung festgehalten.

Die Fragen im Fragebogen beziehen sich auf die räumliche Orientierung in der virtuellen Umgebung, der Natürlichkeit der Bewegung in der virtuellen Umgebung, die Interaktion mit der virtuellen Umgebung, die Erkennbarkeit und Übersichtlichkeit der Visualisierung. Bei jeder dieser Fragen waren die Personen dazu aufgefordert ihre Einschätzung auf einer siebenstelligen Likert-Skala anzugeben. Abbildung 5-5 gibt eine Übersicht über die durchschnittlich angegebenen Skalenwerte bei den verschiedenen Fragen. Die Abbildung 5-5 zeigt, dass die Usability der Anwendung bezogen auf die verschiedenen Aspekte durchweg positiv bewertet wurde, da die durchschnittliche Bewertung deutlich zum positiven Ende (7) der Skala ausgeprägt ist.

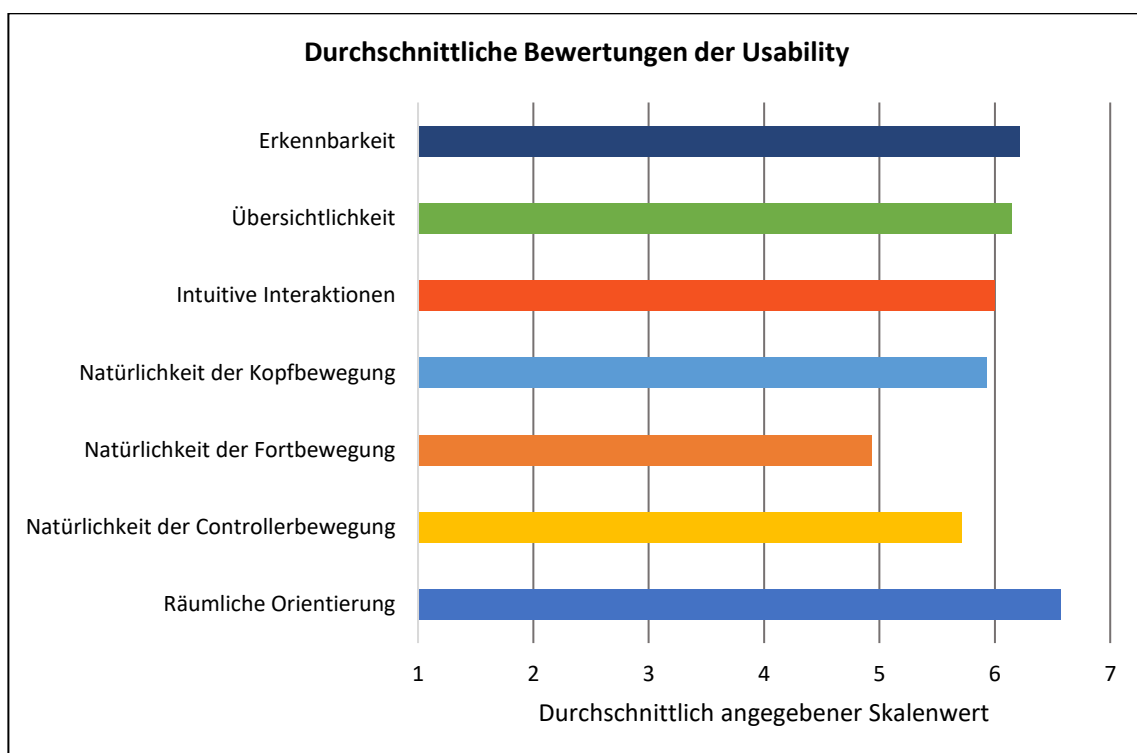


Abbildung 5-5: Übersicht über die durchschnittlichen Bewertungen der Usability

Die räumliche Orientierung in der virtuellen Umgebung wurde von allen Teilnehmern gut bewertet, alle Antworten lagen in der oberen Hälfte der Skala und 64,29 % der Teilnehmer bewerteten sie mit „7: Sehr gut“.

Die Interaktionen mit der virtuellen Umgebung wurden von 92,86 % der Teilnehmer als intuitiv empfunden (sie liegen in der oberen Hälfte der Skala), nur ein Teilnehmer gab den mittleren Wert der Skala an. Die meisten Teilnehmer (50 %) gaben eine Bewertung von 6 ab und 28,57 % empfanden die Interaktionen als sehr intuitiv.

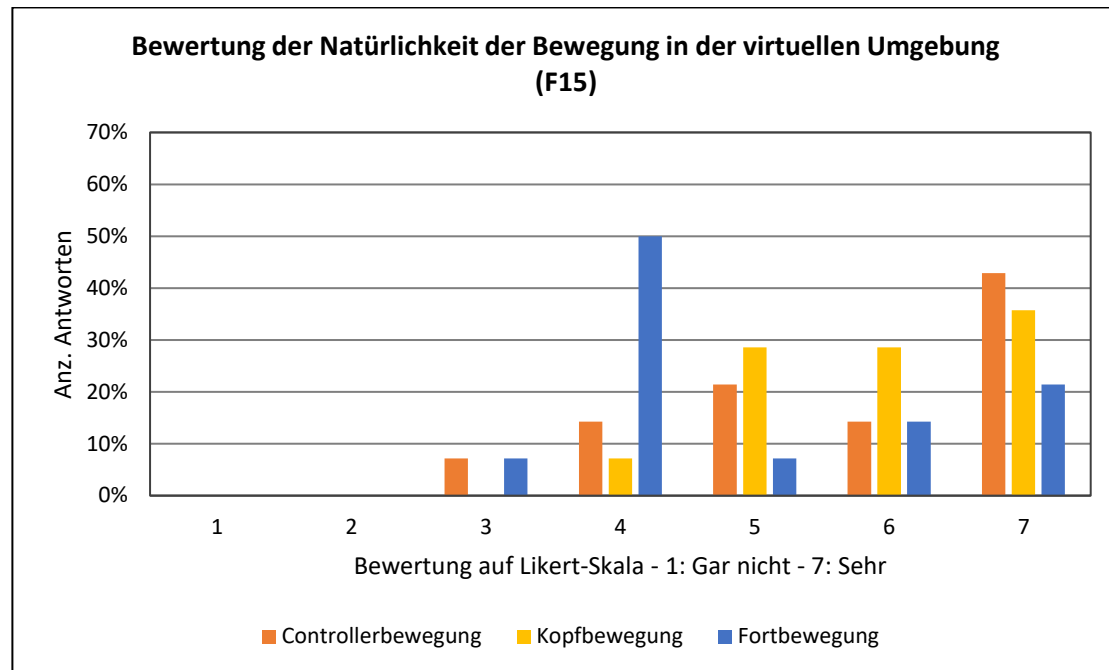


Abbildung 5-6: Bewertung der Natürlichkeit der Bewegung in der virtuellen Umgebung

Bei Frage F15 (siehe Abbildung 5-6), die nach der Natürlichkeit der Bewegung in der virtuellen Umgebung fragt, lagen die Antworten für die Controllerbewegung und die Kopfbewegung ebenfalls fast ausschließlich in der oberen Hälfte der Skala, wobei die Controllerbewegung leicht negativer beurteilt wurde als die Kopfbewegung. Für die Fortbewegung gab die Hälfte der Personen den mittleren Skalenwert und eine Person den Skalenwert 3 an. Die anderen Antworten verteilten sich ebenfalls auf die obere Hälfte der Skala. Die leicht negativere Bewertung der Natürlichkeit der Fortbewegung könnte daran liegen, dass die Anwendung nicht auf viel Fortbewegung ausgelegt ist, und die Personen daher das Gefühl hatten diese nicht ausreichend bewerten zu können. Von einer Person wurde dies auch beim Ausfüllen des Fragebogens geäußert (BO).

Bei der Bewertung der Erkennbarkeit und Übersichtlichkeit der Visualisierung liegen alle Antworten in der oberen Hälfte der Skala, wobei nur zwei Personen (14,29 %) den Skalenwert 5 wählten. Die weiteren 85,71 % der Antworten verteilen sich auf die Skalenwerte 6 und 7, wobei der Skalenwert 6 am häufigsten ausgewählt wurde.

Mit Ausnahme von F15 und einer Antwort bei F16 lagen bei diesen Fragen zur Usability alle Antworten in der oberen Hälfte der Skala und insgesamt gaben die meisten Personen einen Wert von 6 oder 7 an.

Die Frage F17 bezog sich auf das Auftreten von Schwindelgefühlen oder Ähnlichem während der Nutzung der VR-Anwendung. Hier gab nur eine Person eine leichte Desorientierung zu Beginn an.

Bei Frage F19/F19a wurde nach Elementen der Visualisierung gefragt, die schlecht erkennbar waren. Hier gaben drei Personen an, dass es solche Elemente gab. Eine Person gab an, dass Texte etwas unscharf waren, wenn sie nicht im Fokus lagen (FU) und zwei Personen gaben an, dass die kleine Darstellung der Feature-Maps auf den Input- und Output-Leinwänden schlecht erkennbar waren (FM, FH), wobei eine der Personen außerdem ein leichtes Flimmern dieser Elemente erwähnte (FM).

Bei zwei weiteren Fragen (F21a und F27) konnten einzelne Antworten identifiziert werden, die sich auf die Usability der Anwendung beziehen und daher an dieser Stelle genannt werden. Bei der Frage nach Elementen der Anwendung, die das Verständnis nicht unterstützen wurde von einem Teilnehmer angemerkt, dass man den Output „schon bei der Ebene davor“ (F21a, U) sehen kann und dies „überflüssig und redundant“ (F21a, U) erschien. Dies bezieht sich vermutlich auf die Darstellung des Modell-Outputs auf dem Fully-Connected-Layer, die der Darstellung auf dem Output-Layer gleicht. Unter sonstigen Anmerkungen zur Anwendung gab eine Person an, dass der X-Button zum Schließen von Elementen sich häufig an unterschiedlichen Stellen befand und dies unerwartet war (F27, U).

Eine Person äußerte bei der Beobachtung einige positive Kommentare zur Usability der Anwendung, speziell in Bezug auf Virtual Reality. Sie gab an, dass sich durch die Gestaltung der virtuellen Umgebung mit Wänden statt nur eines Bodens und sonst offenem Gelände die Umgebung sehr solide angefühlt hat, im Gegensatz zu anderen VR-Anwendungen, die die Person vorher genutzt hatte (BY). Dadurch hätte man sich präsenter im Raum gefühlt und es traten keine Symptome der Reise- oder Bewegungskrankheit auf (BY). Auch die Leinwände und Modelle hätten sich solide angefühlt und nicht so, als ob sie im Raum schweben würden (BY). Die Leinwände wären außerdem übersichtlich angeordnet gewesen (BY).

Durch die Beobachtung und teilweise den Fragebogen konnten außerdem Möglichkeiten identifiziert werden, die die Usability der Anwendung weiter verbessern könnten und die am wichtigsten erscheinen, da sie sich auf die Ergebnisse von mehreren Teilnehmern

beziehen. Im Fragebogen wurden diese teilweise durch die Frage nach nicht unterstützten bzw. negativ beeinflussenden Elementen (F21/21a) erfasst. Bei der Beobachtung fielen bestimmte Bedienmuster auf oder es wurden entsprechende Äußerungen der Teilnehmer notiert, die hier einfließen. Aus diesen Daten wurden Aussagen gebildet, die als mögliche Verbesserungen der Anwendung formuliert wurden. Bei Überschneidungen wurden die Daten aus den Fragebögen und den Beobachtungen unter den Aussagen zusammengefasst.

Verknüpfung der Detailanzeigen der Input- und Output-Feature-Maps auf dem Pooling-Layer

Bei den Beobachtungen fiel auf, dass auf dem Pooling-Layer häufig versucht wurde, sich die Details von einer Input- und einer Output-Feature-Map mit gleichem Index anzeigen zu lassen, um die Input-Feature-Map und die Output-Feature-Map direkt vergleichen zu können (BE, BG, BL, BS). Teilweise wurde auch im Fragebogen der Wunsch geäußert, dass zwischen den Detailanzeigen der Feature-Maps eine Verknüpfung besteht, wie auch auf dem Convolution-Layer zwischen Filter und Feature-Map (F21a, E; F21a, G).

Entfernung des Schloss-Buttons zur Aufhebung der Verknüpfung der Detailanzeigen zu Filter und Feature-Map

Die Funktion des Schloss-Buttons zur Aufhebung der Verknüpfung der Detailanzeigen zu Filter und Feature-Map auf dem Convolution-Layer schien für die Nutzer häufig nicht eindeutig zu sein. Während der Beobachtungen wurde mehrmals danach gefragt und/oder der Button ohne ersichtlichen Effekt benutzt und danach ignoriert (BU, BG, BA, BZ, BS, BL). Eine Auswirkung des Buttons ist in der Anwendung nur zu sehen, sobald ein anderer Filter oder eine andere Feature-Map ausgewählt wird. Auch im Fragebogen wurde die Frage nach der Funktion des Buttons aufgegriffen (F27, U; F21a, G) und ein Teilnehmer merkte an, dass es besser wäre, wenn Filter und Feature-Map immer verknüpft sind (BG). Daher erscheint es sinnvoll, den Schloss-Button zu entfernen und die Verknüpfung zwischen ausgewähltem Filter und entsprechender Feature-Map immer bestehen zu lassen.

Gemeinsame Übersicht über den Modell-Input und -Output

Während der Nutzung der Anwendung wechselten mehrere Personen häufig zwischen der Modell-Input-Ebene und Modell-Output-Ebene, da sie den Input geändert hatten und das

entsprechende Output-Ergebnis sehen wollten (BE, BS, BY, BL). Hier würde es sich anbieten eine gemeinsame Übersicht über den Modell-Input und -Output zu geben, damit bei der Änderung des Inputs sofort die Auswirkungen auf den Output ersichtlich werden.

Verbesserung der Interaktion mit den Buttons

Wenn die Personen bei einer Interaktion den Controllerbutton drücken, obwohl die Skalierungsanimation des entsprechenden Buttons noch nicht abgeschlossen ist, wird die Interaktion nicht registriert (BW, BO, BA, BZ, BL). Dies schien keine größeren Auswirkungen auf die Bedienung der Anwendung zu haben und wurde von keinem Teilnehmer im Fragebogen erwähnt, für eine Verbesserung der Usability sollte hier die Funktionalität der Interaktion jedoch angepasst werden, etwa durch eine kürzere Animationszeit bei der Skalierung der Buttons.

5.3.5 Erweiterung der Anwendung

Durch die Angaben zu offenen Fragen zu CNNs (F25a), sonstigen Anmerkungen (F27) im Fragebogen und Kommentaren während der Beobachtung konnten einige mögliche Erweiterungen der Anwendung identifiziert werden, welche ihren Funktionsumfang erhöhen könnten. Diese sind im Folgenden aufgeführt.

Als mögliche Erweiterung der Anwendung wurde von zwei Teilnehmern eine Anbindung an ein reales CNN-Modell genannt, um die Parameter des dargestellten Modells interaktiv verändern zu können (F27, M / BM; BQ). In diesem Zusammenhang wurde von einem Teilnehmer die Möglichkeit einer kooperativen VR-Anwendung mit mehreren Personen zum gemeinsamen Erstellen von Netzen als Mehrwert genannt (F27, M). Insgesamt äußerten zwei Teilnehmer den Wunsch, die Anwendung interaktiv zur Entwicklung von Modellen nutzen zu können (F27, M/BM; BY), wobei einer eine Erweiterung zum Debugging hervorhob (BY).

Darüber hinaus wurde die Unterscheidung von gelernten und vom Entwickler festgelegten Parametern des CNNs angeregt (F25a, E) und der Vergleich des CNNs zu klassischen Künstlichen Neuronalen Netzen und anderen Machine-Learning-Verfahren (F27, O). Eine Person sah außerdem eine „Darstellung des Netzes selbst“ (F25a, S) als interessant an und ein Teilnehmer merkte an, dass die Visualisierung „punktuell [...] durch kurze Erklärungstexte ergänzt werden“ (F27, Q) könnte.

5.3.6 Auffälligkeiten in Bezug auf das Vorwissen

Bei der Sichtung der erhobenen Daten fielen einige Auffälligkeiten in Bezug auf das angegebene Vorwissen in Verbindung mit den Antworten/Beobachtungen auf, die in dieser Kategorie aufgeführt werden.

Bei der Sichtung der Daten fiel auf, dass die Teilnehmer, die bei der Beobachtung größere Verständnisprobleme zeigten und denen die Visualisierung ohne weitere Erklärungen keinen Aufschluss über die Funktionsweise des CNNs gab (O, A, W), im Vorfeld zwar angaben im Bereich der Informatik zu arbeiten, aber ein sehr geringes Vorwissen zu allen drei Themenbereichen zu haben. Dabei gaben zwei der drei Personen an, ein geringes Vorwissen im Bereich des Maschinellen Lernens (Skalenwert 2 und 3) zu haben und kein Vorwissen zu Künstliche Neuronale Netze oder CNNs (FA, FO). Eine Person gab in Bezug auf Maschinelles Lernen kein Vorwissen an, aber ein geringes Vorwissen (Skalenwert 2) in Bezug auf Künstliche Neuronale Netze und CNNs (FW).

Im Gegensatz dazu äußerte sich ein Teilnehmer (L), der nicht im Bereich der Informatik arbeitet und kein Vorwissen zu den Themen hatte, sehr positiv zu der Anwendung. Er gab nach der Nutzung der Anwendung an, dass er es „super spannend“ fand und äußerte während der Nutzung der Anwendung Kommentare, die auf ein Verständnis der Vorgänge des CNNs schließen lassen (BL). So wurde beispielsweise die Verbindung von Input und Output zwischen den Ebenen erkannt und analysiert, welche Inputbilder gut erkannt werden und welche nicht (BL).

Die Personen, die den Wunsch nach einer Verbindung zu einem CNN-Modell äußerten und/oder sich die Anwendung als Entwicklungsumgebung vorstellen könnten, gehörten zu den Personen mit einer Angabe des Vorwissen von 4 oder höher im Bereich des Maschinellen Lernens und der Künstlichen Neuronale Netze (M, Q) und/oder gaben an im Bereich des Maschinellen Lernens zu arbeiten (Q, Y).

5.4 Diskussion & Auswertung

Durch die Abfrage des Wissensstandes zu Beginn der Anwendung, nach dem Lesen eines Einführungstextes und nach der Nutzung der Anwendung gaben 78,57% der Teilnehmer eine **Wissenssteigerung** durch die Nutzung der Anwendung an. Im Rahmen dieser Evaluierung kann jedoch nicht festgestellt werden, ob dieser Wissenszuwachs vor allem auf das

Visualisierungskonzept und den Einsatz von VR zurückzuführen ist, oder allgemein darauf, dass sich die Teilnehmer durch die Anwendung einige Minuten mit dem Thema auseinandergesetzt haben.

Drei der Hauptaspekte des **Visualisierungskonzeptes** sind die Visualisierung der Filter und Feature-Maps, die Gegenüberstellung von Input und Output der Ebenen und die Übersicht über das komplette Convolutional Neural Network durch das 3D-Modell. Diese Aspekte werden durch die Ergebnisse der Evaluierung unterstützt, da sie von mehreren Teilnehmern als besonders unterstützende Elemente der Visualisierung genannt wurden.

Durch die Evaluierung konnten jedoch auch Aspekte in Bezug auf den Wissenserwerb durch die Visualisierung festgestellt werden, die in nachfolgenden Versionen verbessert werden könnten. Die häufigste offene Frage zur Visualisierung und CNNs bezieht sich auf die Bedeutung der Parameter im Detailfenster des kleinen 3D-Modells, wie die Angabe von Stride und Padding des Convolutional-Layers. Es erscheint somit sinnvoll, dieses Element der Visualisierung durch Erläuterungen zu ergänzen. Die Frage mit den zweithäufigsten Nennungen bezieht sich auf den Lernprozess des CNNs, der durch die aktuelle Visualisierung nicht erläutert wird. Ein Nutzer schlug als Erweiterung der Anwendung die Markierung von gelernten und vom Entwickler festgelegten Parametern vor, was ein erster Schritt zur Erläuterung des Lernprozesses in der Visualisierung sein könnte.

Auffällig war, dass es teilweise sehr individuelle Ausprägungen in der Beurteilung der Anwendung gab, teilweise auch unabhängig vom angegebenen Vorwissen. Für einige Nutzer mit sehr geringem oder keinem Vorwissen zum Thema Maschinelles Lernen schien die Visualisierung ohne zusätzliche Erklärungen teilweise nicht verständlich zu sein. Dies unterstützt die These, dass die Anwendung nur als Zusatz, nicht aber als alleiniges Lernmedium genutzt werden kann. Außerdem ist die Anwendung auf die Darstellung von CNNs fokussiert, die bereits einem sehr spezifischen Bereich des Maschinellen Lernens und der Künstlichen Neuronalen Netze entsprechen und somit nicht als Einstieg in das Oberthema geeignet erscheinen. Der Einführungstext konnte dabei nur bedingt den richtigen Einstieg in die drei aufeinander aufbauenden Themen gewährleisten. Bei Nutzern mit mehr Erfahrung in dem Bereich des Maschinellen Lernens kam die Visualisierung hingegen gut an, was diese These unterstützt.

Generelle Aussagen zum Nutzerverhalten in Verbindung mit dem Vorwissen zu treffen erscheint jedoch schwierig, da sich ein Teilnehmer ohne Vorwissen und ohne Informatikbezug sehr positiv bezüglich der Anwendung und seinem Wissenszuwachs geäußert hat.

Die Anwendung scheint daher für eine **Nutzergruppe** geeignet zu sein, die sich im Vorfeld bereits teilweise mit dem Thema des Maschinellen Lernens und eventuell auch mit Künstlichen Neuronalen Netzen beschäftigt hat und durch die Anwendung einen ersten Einblick in die Funktionsweise eines CNNs bekommen möchte. Hier wäre jedoch eine weitere Untersuchung mit Nutzergruppen mit unterschiedlichem Vorwissen denkbar, um näher zu spezifizieren für welche Nutzergruppe sich die Anwendung besonders eignet, oder welche Änderungen für eine bestimmte Zielgruppe vorgenommen werden müssten. Für Einsteiger in das Thema müssten der Visualisierung anscheinend noch zusätzliche Informationen, wie etwa durch die von einem Teilnehmer genannten Erklärungstexte, hinzugefügt werden.

Das Visualisierungskonzept beinhaltet die Hypothese, dass sich die Nutzer durch den **Einsatz von Virtual Reality** besser auf die dargestellten Daten konzentrieren können, da äußere Ablenkung reduziert werden. Dies wird durch die Ergebnisse der Evaluierung unterstützt, da es sowohl bei der Begründung zur Einschätzung zum Einsatz von VR genannt wurde als auch auf der Likert-Skala entsprechend hoch bewertet wurde. Des Weiteren wurde von einem Teilnehmer angegeben, dass die Motivation sich mit dem Thema CNN zu befassen durch den Einsatz der innovativen VR-Technologie erhöht wurde und die Interaktion dadurch eine spielerische Komponente bekam.

In Bezug auf den Einsatz von Virtual Reality gibt es aber auch Verbesserungspotential bei der Anwendung, da die Hälfte Teilnehmer angab, sich die Visualisierung auch gut an einem PC-Bildschirm als 2D- oder 3D-Visualisierung vorstellen zu können. Hier scheint es insbesondere bei der Raumnutzung durch die Visualisierung Verbesserungspotential zu geben, da drei der Teilnehmer diese als unzureichend empfunden haben, um den Einsatz von Virtual Reality zu rechtfertigen. Dies könnte ein Ansatz sein, um die Vorteile von Virtual Reality in der Anwendung deutlicher auszunutzen. Auch beim aktuellen Stand der Anwendung gaben drei Teilnehmer bereits an, dass die Raumnutzung der Visualisierung die Darstellung unterstützt hat. Eine Ausweitung der Raumnutzung könnte daher ein Ansatzpunkt bei einer Weiterentwicklung der Anwendung sein.

Obwohl eine Vielzahl der Teilnehmer eine höhere Konzentration auf die Visualisierung im Vergleich zu PC-/Laptop-Bildschirmen angab, gab die Hälfte der Teilnehmer an, sich die Anwendung auch gut als 2D- oder 3D-Anwendung vorstellen zu können und die Bewertung des Einsatzes von VR fiel leicht schlechter aus, als die Bewertung der höheren Konzentration. Somit scheint die höhere Konzentration auf die Visualisierung für einige Teilnehmer kein ausschlaggebendes Argument für den Einsatz von VR zu sein.

Für die Verwendung von VR-Anwendungen bestehen im Moment noch Einschränkungen, die sich beispielsweise durch die benötigte Hardware ergeben. Ob sich der Einsatz von Virtual Reality für die Anwendung lohnt, vor allem im Hinblick auf diese Einschränkungen, könnte in einer Folgestudie mit einem direkten Vergleich zu einer ähnlichen 2D- oder 3D-Anwendung am PC-Bildschirm evaluiert werden.

Die **Usability** der Anwendung wurde von den Nutzern insgesamt gut bewertet, vor allem im Hinblick auf die Erkennbarkeit und Übersichtlichkeit. Die Wahl der **Interaktionsmethode** des Raycast-Pointers wird dadurch unterstützt, dass die Interaktionen mit der virtuellen Umgebung von einer großen Mehrheit der Teilnehmer als intuitiv bewertet wurde. Vor allem durch die Beobachtungen konnten kleinere Verbesserungsmöglichkeiten für die Bedienung der Anwendung identifiziert werden, die bei einer Weiterentwicklung berücksichtigt werden sollten (siehe Abschnitt 5.3.4, S.61).

In Bezug auf **mögliche Erweiterungen der Anwendung** durch neue Funktionen oder Ergänzungen scheint vor allem die im Konzept bereits angedachte Anbindung an ein CNN-Modell, wodurch die Änderung von Parametern des Modells live in der Anwendung ermöglicht wird, durch die Evaluierungsergebnisse unterstützt zu werden. Dies wird bestärkt durch die Fragen der Teilnehmer nach den einstellbaren Parametern und den Auswirkungen von veränderten Parametern, die durch eine solche Ergänzung beantwortet werden könnten. Von den Personen, die das höchste Vorwissen in den drei Themenbereichen angaben bzw. im Bereich des Maschinellen Lernens arbeiten, wird eine Nutzung der Anwendung außerhalb des Einsatzes als Lernumgebung als sinnvoll erachtet, genannt wurde die Nutzung zum Entwickeln und Debuggen von CNNs. Bei der Weiterentwicklung der Anwendung könnte dementsprechend auch dieses Nutzerszenario betrachtet und evaluiert werden.

Einschränkungen

Insgesamt kann diese Evaluierung durch den stark eingegrenzten Teilnehmerkreis und den qualitativen Ansatz nur Tendenzen aufzeigen, welche die Hypothesen zum Konzept und der Anwendung unterstützen oder nicht, aber keine repräsentativen Ergebnisse liefern.

Die angegebenen Erfahrungen in den drei Themenbereichen sind nur subjektive Einschätzungen der Teilnehmer und wurden nicht durch einen Wissenstest oder ähnliches bestätigt. Bei der Evaluierung ging es primär darum, den gefühlten Wissensfortschritt der Teilnehmer zu erkennen.

Zu beachten ist außerdem, dass die Teilnehmer im Schnitt 8,6 Minuten in der VR-Umgebung verbracht haben. Die kürzeste Verweildauer in der Anwendung betrug 4:03 Minuten und die Längste 16:25 Minuten. Dies erscheint wenig Zeit zu sein, um sich mit einem komplexen Thema wie CNNs auseinanderzusetzen. Durch einen längeren Aufenthalt in der Anwendung im Rahmen einer ausführlicheren Evaluierung könnte sich der Wissenszuwachs durch die Anwendung weiter steigern. Aufgrund dieser erheblichen Zeitunterschiede in der Verweildauer in der Anwendung sollten bei zukünftigen Evaluierungen die Gründe für diese Unterschiede erfragt werden.

Zu den Methoden

Durch die Beobachtungen konnten vor allem Auffälligkeiten in der Bedienung der Anwendung festgehalten werden, wodurch Verbesserungsmöglichkeiten identifiziert werden konnten.

Bei der Auswertung des Fragebogens fiel auf, dass manche Teilnehmer bei der Einschätzung der Veränderung des Wissensstandes Schwierigkeiten hatten, da sie nicht mehr genau wussten, was sie zuvor beim Wissensstand angegeben hatten. Hier könnte man den zuvor angegebenen Wissensstand erneut zum Vergleich anzeigen oder nur die Veränderung abfragen. Bei den offenen Fragen im Fragebogen könnte es sich anbieten, diese in einer mündlichen Befragung zu erheben, da dort die Möglichkeit zu Nachfragen gegeben ist, wodurch unklare Äußerungen direkt spezifiziert werden könnten.

6 Fazit & Ausblick

Im Rahmen dieser Arbeit wurde ein CNN in VR visualisiert. Die Arbeit unterteilte sich in drei Schritte, die Entwicklung eines Konzepts, die Umsetzung als Prototyp und die anschließende Evaluierung.

Es sollte eine VR-Anwendung konzipiert werden, die mehrere Anforderungen erfüllen musste. Die Anwendung sollte weitestgehend selbsterklärend sein und trotzdem die komplexen Strukturen eines CNNs darstellen. Die Konzeptionsphase gestaltete sich zu Anfang schwieriger und aufwendiger als gedacht. Eine Herausforderung war dabei die notwendige Datenreduzierung unter Beibehaltung des Erkenntnisfortschritts, den die Visualisierung liefern sollte. Die Reduzierung der darzustellenden Datenmenge muss von Anfang an konzeptionell berücksichtigt werden. Die Zusammenfassung der Daten in ein vereinfachtes 3D-Modell zur ersten Übersicht über die Abläufe im CNN hat sich in der Evaluierung bewährt. Der Aufbau der Visualisierung, insbesondere durch die Darstellung des Ablaufs im CNN-Modell durch das 3D-Modell, wurde bei der Evaluierung positiv hervorgehoben, ebenso wie die Darstellung der Filter und Feature-Maps und die Interaktion mit der Visualisierung. Die Ergebnisse der Evaluierung unterstützen außerdem die These, dass die Immersion der VR-Anwendung die Konzentration auf die dargestellten Daten fördert. Als weiterer positiver Effekt der VR-Technologie wurde eine erhöhte Motivation, sich mit dem Thema CNNs zu befassen, festgestellt. Bei der Evaluierung wurde ein subjektiv empfundener Wissenszuwachs zu der Funktionsweise von CNNs durch die Visualisierung in VR festgestellt, was die Möglichkeit zur Nutzung der Anwendung als Lernumgebung bekräftigt.

Aus dem Gestaltungsprozess und der Evaluierung können mehrere Erkenntnisse auch für zukünftige Projekte und Anwendungen gezogen werden. In Bezug auf die Raumnutzung in VR wurde bei der Konzeption der Visualisierung darauf geachtet, sie möglichst übersichtlich zu gestalten. Dabei entstand ein Konzept, bei dem die einzelnen Elemente relativ nah beieinander liegen und besser aus etwas Entfernung, anstatt aus direkter Nähe betrachtet werden können. Dadurch sollten möglichst viele Informationen gleichzeitig wahrgenommen werden können, auch ohne sich viel bewegen oder seine Perspektive ändern zu müssen. Die Evaluierung hat jedoch gezeigt, dass bei der Verwendung einer VR-Anwendung gerade diese Aspekte der Bewegung und Änderung der Perspektive in der virtuellen

Umgebung erwartet werden. Daher wäre es ein Ansatz für weitere Arbeiten den Aspekt der Raumnutzung im Visualisierungskonzept weiter auszubauen und sinnvoll zu integrieren. Eine Möglichkeit wäre es dabei, die einzelnen Visualisierungselemente bewegbar zu machen, sodass sich der Nutzer den Aufbau der Visualisierung im Raum zu einem gewissen Grad selbst nach eigenen Bedürfnissen und Präferenzen gestalten kann. Dabei wäre jedoch darauf zu achten, dass der Einstieg in die Visualisierung dadurch nicht zu sehr verzögert wird und man sich auf wenige, aber sinnvolle Einstellungsmöglichkeiten beschränkt. Auch könnte eine Evaluierung mit einer Anwendung durchgeführt werden, die diverse Einstellungsmöglichkeiten bietet, um zu eruieren welche besonders häufig genutzt werden und ob sich Muster bei der Platzierung der Elemente im Raum ergeben. Als weiterer Ansatz könnte die Nutzung von 3D-Elementen in der Visualisierung ausgeweitet werden. Denkbar wäre die Ergänzung des Fully-Connected-Layer durch ein 3D-Modell, welches die Neuronen und Verbindungen zumindest im Ansatz darstellt, statt der 2D-Darstellung auf der Leinwand.

Dieses Konzept wurde unter dem Use-Case einer Lernumgebung betrachtet, um den Einstieg in den Themenbereich der CNNs zu begleiten. Die Evaluierung hat gezeigt, dass für das Konzept auch andere Use-Cases denkbar sind, wie der Einsatz als Entwicklungsumgebung und zum kooperativen Erstellen von Modellen. Auch hier könnten weitere Arbeiten ansetzen.

Anschließend an diese Arbeit könnte der Prototyp erweitert werden, etwa durch die Anbindung an ein CNN-Modell, womit auch die Interaktivität der Anwendung erhöht werden könnte, indem man beispielsweise das Hinzufügen von Layern und das Ändern von Parametern einzelner Layer, wie der Filtergröße, ermöglicht. Des Weiteren könnte die Einbindung von mehreren CNN-Modellen umgesetzt werden, um einen Vergleich der Inputverarbeitung von verschiedenen aufgebauten Modellen zu ermöglichen.

Um den Einfluss von VR bei dieser Visualisierung weiter zu untersuchen, sollte eine Evaluierung angeschlossen werden, wobei der Prototyp im direkten Vergleich mit einer ähnlich strukturierten Anwendung am PC-Bildschirm evaluiert wird.

Literaturverzeichnis

- Abulrub, A. G., Attridge, A. N., & Williams, M. A. (2011). Virtual Reality in engineering education: The future fo creative learning. *2011 IEEE Global Engineering Education Conference (EDUCON)*, (S. 751-757). Amman, Jordan. <https://doi.org/10.1109/EDUCON.2011.5773223>
- Bortz, J., & Döring, N. (2006). *Forschungsmethoden und Evaluation* (4. Ausg.). Springer Medizin Verlag Heidelberg.
- Broll, W. (2013). Augmentierte Realität. In R. Dörner, W. Broll, P. Grimm & B. Jung (Hrsg.), *Virtual und Augmented Reality (VR/AR): Grundlagen und Methoden der Virtuellen und Augmentierten Realität* (S. 241-294). Springer-Verlag Berlin Heidelberg. https://doi.org/10.1007/978-3-642-28903-3_8
- Burton, S. (20. April 2018). Making the case for safety of Machine Learning applied to highly automated driving. *FoMLAS'18*. Thessaloniki, GRC. Abgerufen am 21. Juli 2019 von https://download.fortiss.org/public/FoMLAS/2018/FoMLAS18_Keynote_Simon_Burton.pdf
- Chollet, F. (2018). *Deep Learning mit Python und Keras*. Frechen: mitp Verlag.
- Chu, G., Humer, I., & Eckhardt, C. (2019). Special Relativity in Immersive Learning. In D. Beck et al. (Hrsg.), *Immersive Learning Research Network*, (S. 16-29). https://doi.org/10.1007/978-3-030-23089-0_2
- Dede, C., Salzman, M. C., & Loftin, R. B. (1996). ScienceSpace: Virtual realities for learning complex and abstract scientific concepts. *Proceedings of the IEE 1996 Virtual Reality Annual International Symposium*, (S. 246-252). Santa Clara, CA, USA. <https://doi.org/10.1109/VRAIS.1996.490534>
- Deutsches Zentrum für Luft- und Raumfahrt e. V. (o.D.-a). *DLR: Simulations- und Softwaretechnik*. Abgerufen am 12. Juli 2019 von https://www.dlr.de/sc/desktopdefault.aspx/tabid-1185/1634_read-3062/
- Deutsches Zentrum für Luft- und Raumfahrt e. V. (o.D.-c). *DLR: Simulations- und Softwaretechnik - Intelligente und Verteilte Systeme*. Abgerufen am 12. Juli 2019 von https://www.dlr.de/sc/desktopdefault.aspx/tabid-1199/1657_read-3066/

- Deutsches Zentrum für Luft- und Raumfahrt e.V. (o.D.-b). *DLR: Simulations- und Softwaretechnik - Home*. Abgerufen am 12. Juli 2019 von <https://www.dlr.de/sc>
- Döring, N. & Bortz, J. (2016). *Forschungsmethoden und Evaluation* (5. Ausg.). Springer-Verlag Berlin Heidelberg.
- Dörner, R., Jung, B., Grimm, P., Broll, W., & Göbel, M. (2013). Einleitung. In R. Dörner, W. Broll, P. Grimm, & B. Jung (Hrsg.), *Virtual und Augmented Reality (VR/AR): Grundlagen und Methoden der Virtuellen und Augmentierten Realität* (S. 1-31). Berlin Heidelberg: Springer-Verlag.
- Facebook Technologies LLC. (o.D.-a). *Oculus Quest: Features*. Abgerufen am 12. Juli 2019 von <https://www.oculus.com/quest/features/>
- Facebook Technologies LLC. (o.D.-b). *Oculus Rift S: Features*. Abgerufen am 12. Juli 2019 von <https://www.oculus.com/rift-s/features/>
- Fauster, R. (15. März 2016). *PlayStation VR: Preis und Veröffentlichungsdatum enthüllt*. Abgerufen am 12. Juli 2019 von <https://blog.de.playstation.com/2016/03/15/playstation-vr-preis-und-verffentlichungsfenster-enthllt/>
- Gesellschaft für Informatik e. V. (2018). *Stellungnahme der Gesellschaft für Informatik e. V. (GI) zur Nationalen Strategie für Künstliche Intelligenz der Bundesregierung*. Abgerufen am 21. Juli 2019 von https://gi.de/fileadmin/GI/Allgemein/PDF/2018-12-03_Stellungnahme_KI_Strategie.pdf
- Google Developers. (15. Februar 2017). *Hands-on TensorBoard (TensorFlow Dev Summit 2017)*. Abgerufen am 12. Juli 2019 von <https://www.youtube.com/watch?v=eBbEDRsCmv4>
- Google LLC. (o.D.-a). *Daydream View*. Abgerufen am 12. Juli 2019 von https://vr.google.com/intl/de_de/daydream/smartphonevr/
- Google LLC. (o.D.-b). *TensorBoard overview*. Abgerufen am 12. Juli 2019 von <https://www.tensorflow.org/tensorboard/r1/overview>
- Google LLC. (o.D.-c). *TensorBoard: Visualizing Learning*. Abgerufen am 12. Juli 2019 von <https://www.tensorflow.org/tensorboard/r1/summaries>

- Harley, A. W. (2015). An interactive Node-Link visualization of Convolutional Neural Networks. *ISVC 2015, Part I*, (S. 867-877). Las Vegas, NV, USA. https://doi.org/10.1007/978-3-319-27857-5_77
- Harley, A. W. (o.D.). *3D convolutional network visualization*. Abgerufen am 8. Juli 2019 von <http://www.cs.cmu.edu/~aharley/vis/conv/>
- Heise Online. (o.D.). *HTC Vive: VR-Brille mit Holodeck*. Abgerufen am 12. Juli 2019 von <https://www.heise.de/thema/HTC-Vive>
- Hope, T., Resheff, S. Y., & Lieder, I. (2018). *Einführung in TensorFlow: Deep-Learning-Systeme programmieren, trainieren, skalieren und deployen*. Heidelberg: dpunkt.verlag.
- HTC Corporation. (2019a). *Vive Pro VR System*. Abgerufen am 4. Juli 2019 von <https://www.vive.com/de/product/vive-pro-full-kit/>
- HTC Corporation. (2019b). *Vive VR-System*. Abgerufen am 4. Juli 2019 von <https://www.vive.com/de/product/>
- Irvine, M. A. (6. April 2017). *Introduction to Convolutional Neural Networks*. Abgerufen am 24. Juli 2019 von https://sempwn.github.io/blog/2017/04/06/conv_net_intro
- IT Center der RWTH Aachen. (9. Mai 2019). *aixCAVE der RWTH Aachen*. Abgerufen am 3. Juli 2019 von <http://www.itc.rwth-aachen.de/cms/IT-Center/Forschungs-Projekte/Virtuelle-Realitaet/Infrastruktur/~fgqa/aixCAVE/>
- Janssen, J.-K. (29. März 2016). Oculus Rift im Test: Virtual Reality für die Massen. *c't magazin für computer technik*. Abgerufen am 9. Juli 2019 von <https://www.heise.de/ct/artikel/Oculus-Rift-im-Test-Virtual-Reality-fuer-die-Massen-3151909.html>
- Janssen, J.-K. (27. Mai 2019). *Oculus Quest: High-End-VR ohne PC oder Konsole*. Abgerufen am 12. Juli 2019 von <https://www.heise.de/tests/Oculus-Quest-High-End-VR-ohne-PC-oder-Konsole-4430044.html>
- Karpathy, A. (o.D.-a). *ConvNetJS: Deep Learning in your browser*. Abgerufen am 12. Juli 2019 von <https://cs.stanford.edu/people/karpathy/convnetjs/>

- Karpathy, A. (o.D.-b). *ConvNetJS MNIST demo*. Abgerufen am 12. Juli 2019 von <https://cs.stanford.edu/people/karpathy/convnetjs/demo/mnist.html>
- Keras-Team. (22. Februar 2018). *keras/mnist_cnn.py*. Abgerufen am 17. Juli 2019 von https://github.com/keras-team/keras/blob/master/examples/mnist_cnn.py
- Keras-Team. (o.D.-a). *Keras Documentation: Datasets*. Abgerufen am 5. Juli 2019 von <https://keras.io/datasets/>
- Keras-Team. (o.D.-b). *Keras Documentation: Home*. Abgerufen am 20. Juni 2019 von <https://keras.io/>
- Knight, W. (11. April 2017). The dark secret at the heart of AI. *MIT Technology Review*. Abgerufen am 9. Januar 2019 von <https://www.technologyreview.com/s/604087/the-dark-secret-at-the-heart-of-ai/>
- LeCun, Y., Bengio, Y., & Hinton, G. (28. Mai 2015). Deep learning. *Nature*, S. 436-444. <https://doi.org/10.1038/nature14539>. Aufgerufen am 21. Juli 2019 von https://creativecoding.soe.ucsc.edu/courses/cs523/slides/week3/DeepLearning_LeCun.pdf
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning applied to document recognition. *Proceedings of the IEEE*, (S. 2278 - 2324). <https://doi.org/10.1109/5.726791>
- Likert, R. (1932). A technique for the measurement of attitudes. (R. S. Woodworth, Hrsg.) *Archives of Psychology*, 22(140), S. 5-55. Abgerufen am 17. Juli 2019 von https://legacy.voteview.com/pdf/Likert_1932.pdf
- Lindner, R. (26. März 2014). *Facebook kauft Oculus: Milliarden für Visionen*. Abgerufen am 12. Juli 2019 von <https://www.faz.net/aktuell/wirtschaft/netzwirtschaft/milliarden-deal-facebook-kauft-oculus-12865286.html>
- Mainzer, K. (2016). *Künstliche Intelligenz: Wann übernehmen die Maschinen?* Springer-Verlag Berlin Heidelberg.
- McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (31. August 1955). A proposal for the Dartmouth summer research project on Artificial Intelligence.

- Abgerufen am 12. Juli 2019 von <http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html>
- Mintz, R., Litvak, S., & Yair, Y. (2001). 3D-Virtual Reality in science education: An implication for astronomy teaching. *Journal of Computers in Mathematics and Science Teaching*, 20(3), S. 293-305. Abgerufen am 9. Juli 2019 von <https://www.learntechlib.org/p/9543/>
- Psothka, J. (November 1995). Immersive training systems: Virtual reality and education and training. *Instructional Science*, 23(5-6), S. 405-431. <https://doi.org/10.1007/BF00896880>
- Rashid, T. (2017). *Neuronale Netze selbst programmieren: Ein verständlicher Einstieg mit Python*. Heidelberg: dpunkt.verlag.
- Robertson, A. (30. März 2019). *Oculus Quest Review: A great system with frustrating compromise*. Abgerufen am 12. Juli 2019 von <https://www.theverge.com/2019/4/30/18523000/oculus-quest-review-vr-headset-price-specs-features>
- Schalkwyk, J. (12. März 2019). *An All-Neural On-Device Speech Recognizer*. Abgerufen am 12. Juli 2019 von <https://ai.googleblog.com/2019/03/an-all-neural-on-device-speech.html>
- Shih, J. (4. April 2019). *Using TensorFlowSharp in Unity*. Abgerufen am 15. Juli 2019 von <https://github.com/Unity-Technologies/ml-agents/blob/master/docs/Using-TensorFlow-Sharp-in-Unity.md>
- Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. *Proceedings 1996 IEEE Symposium of Visual Languages*, (S. 336-343). Boulder, CO, USA. <https://doi.org/10.1109/VL.1996.545307>
- Shneiderman, B. (o.D.). *The eight golden rules of interface design*. Abgerufen am 21. Juli 2019 von <https://www.cs.umd.edu/users/ben/goldenrules.html>
- Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., & Hassabis, D. (19. Oktober 2017). Mastering the game of Go without human knowledge. *Nature*, S. 354-359. Abgerufen am 11. Juli 2019 von <https://www.nature.com/articles/nature24270>

- Siri Team. (August 2017). *Improving Neural Network Acoustic Models by Cross-bandwidth and Cross-lingual Initialization*. Abgerufen am 12. Juli 2019 von <https://machinelearning.apple.com/2017/08/01/cross-initialization.html>
- Sony Interactive Entertainment Europe Limited. (2019). *Technische Spezifikationen von PS VR*. Abgerufen am 12. Juli 2019 von <https://www.playstation.com/de-de/explore/playstation-vr/tech-specs/>
- Statt, N. (16. April 2019). *Epic vs. Steam: the console war reimagined on the PC*. (T. Verge, Herausgeber) Abgerufen am 14. Juli 2019 von <https://www.theverge.com/2019/4/16/18334865/epic-games-store-versus-steam-valve-pc-gaming-console-war-reimagined>
- Stöcker, C. (29. Oktober 2017). Künstliche Intelligenz: Ein Gott braucht keinen Lehrmeister. *Spiegel Online*. Abgerufen am 11. Juli 2019 von <https://www.spiegel.de/wissenschaft/technik/kuenstliche-intelligenz-gott-braucht-keine-lehrmeister-kolumne-a-1175130.html>
- Unity Technologies. (2019a). *Unity Download-Archiv*. Abgerufen am 3. Juli 2019 von <https://unity3d.com/de/get-unity/download/archive>
- Unity Technologies. (2019b). *Products: Unity*. Abgerufen am 3. Juli 2019 von <https://unity3d.com/unity>
- Unity Technologies. (2019c). *Unity User Manual (2018.4): Scenes*. Abgerufen am 5. Juli 2019 von <https://docs.unity3d.com/2018.4/Documentation/Manual/Creating Scenes.html>
- Unity Technologies. (21. Februar 2019d). *Unity Inference Engine*. Abgerufen am 15. Juli 2019 von <https://github.com/Unity-Technologies/ml-agents/blob/master/docs/Unity-Inference-Engine.md>
- Valve Corporation. (2019a). *Steam VR*. Abgerufen am 4. Juli 2019 von <https://www.steamvr.com/en/>
- Valve Corporation. (2019b). *Steam VR Plugin*. Abgerufen am 4. Juli 2019 von <https://assetstore.unity.com/packages/tools/integration/steamvr-plugin-32647>

- Valve Corporation. (2019c). *Steam: Die ultimative Online Gaming Plattform*. Abgerufen am 14. Juli 2019 von <https://store.steampowered.com/about/>
- Valve Corporation. (o.D.). *SteamVR Unity Plugin*. Abgerufen am 5. Juli 2019 von https://valvesoftware.github.io/steamvr_unity_plugin/
- von Zitzewitz, G. (2017). *Survey of neural networks in autonomous driving*. Abgerufen am 12. Juli 2019 von https://www.researchgate.net/publication/324476862_Survey_of_neural_networks_in_autonomous_driving
- VR with Andrew (Andrew Connell). (27. Februar 2019a). [01][Unity] *SteamVR Canvas Pointer*. Abgerufen am 12. Juli 2019 von <https://www.youtube.com/watch?v=3mRI1hu9Y3w>
- VR with Andrew (Andrew Connell). (6. März 2019b). [02][Unity] *SteamVR Canvas Pointer*. Abgerufen am 12. Juli 2019 von https://www.youtube.com/watch?v=h_BMXDWv10I
- Witmer, B. G., & Singer, M. J. (Juni 1998). Measuring presence in virtual environments: A presence questionnaire. *Presence*, 7(3), S. 225-240. Abgerufen am 4. Juli 2019 von <https://nil.cs.uno.edu/publications/papers/witmer1998measuring.pdf>
- Zobel, B., Werning, S., Metzger, D., & Thomas, O. (2018). Augmented und Virtual Reality: Stand der Technik, Nutzenpotenziale und Einsatzgebiete. In C. de Witt, & C. Gloerfeld, *Handbuch Mobile Learning* (S. 123-140). Wiesbaden: Springer VS. https://doi.org/10.1007/978-3-658-19123-8_7

Anhang

Inhalte auf beigefügter CD

- A** Bildschirmaufnahmen des Prototyps
- B** Durch den Fragebogen erhobene Daten
- C** Durch die Beobachtung erhobene Daten
- D** Einführungstext der Evaluierung
- E** Datenschutzerklärung der Evaluierung
- F** Unity-Projekt des Prototyps
- G** Windows-Export-Datei des Prototyps
- H** Python-Skripte zur Erzeugung der Beispieldaten

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Masterarbeit selbständig und ohne unzulässige fremde Hilfe angefertigt habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und alle Ausführungen, welche anderen Quellen wörtlich oder sinngemäß entnommen wurden, entsprechend kenntlich gemacht. Diese Arbeit wurde weder in gleicher noch in ähnlicher Form einem anderen Prüfungsamt vorgelegt oder veröffentlicht.

Ort, Datum

Unterschrift

Kontaktinformationen

Nadine Meißler

Am Lilienvveen 87a

46485 Wesel

nadine.meissler@gmx.de